DESIGN OF COUPLED MACE FILTERS FOR OPTICAL PATTERN RECOGNITION USING PRACTICAL SPATIAL LIGHT MODULATORS

Final Report
Contract No. NAG 9-614, Basic

Submitted to:
Lyndon B. Johnson Space Center
National Aeronautics and Space Administration
Houston, Texas

By:
Dr. P. K. Rajan
and
Ajmal Khan

June 1993

Center for Manufacturing Research
and
Technology Utilization

Cookeville, Tennessee 38505

DESIGN OF COUPLED MACE FILTERS FOR OPTICAL PATTERN RECOGNITION USING PRACTICAL SPATIAL LIGHT MODULATORS

Final Report

Contract No. NAG 9-614, Basic

Submitted to:

Lyndon B. Johnson Space Center
National Aeronautics and Space Administration
Houston, Texas

By:

Dr. P. K. Rajan
and
Ajmal Khan

Center for Manufacturing Research
and Technology Utilization
Tennessee Technological University
Cookeville, TN 38505

June 1993

I

ABSTRACT

Spatial light modulators (SLMs) are being used in correlation-based optical pattern recognition systems to implement the Fourier domain filters. Currently available SLMs have certain limitations with respect to the realizability of these filters. Therefore it is necessary to incorporate the SLM constraints in the design of the filters.

The design of a SLM-constrained minimum average correlation energy (SLM-MACE) filter using the simulated annealing-based optimization technique was investigated. The SLM-MACE filter was synthesized for three different types of constraints. The performance of the filter was evaluated in terms of its recognition (discrimination) capabilities using computer simulations. The correlation plane characteristics of the SLM-MACE filter were found to be reasonably good. The SLM-MACE filter yielded far better results than the analytical MACE filter implemented on practical SLMs using the constrained magnitude technique.

Further, the filter performance was evaluated in the presence of noise in the input test images. This work demonstrated the need to include the SLM constraints in the filter design. Finally, a method is suggested to reduce the computation time required for the synthesis of the SLM-MACE filter.

ACKNOWLEDGEMENTS

The authors would like to express their appreciation to Dr. Richard D. Juday of NASA Johnson Space Center and Dr. Stanley E. Monroe, Jr., of Lockheed Engineering and Sciences Corporation for their consultations and advice given during the project.

The authors would like to thank the Lyndon B. Johnson Space Center, National Aeronautical and Space Administration (NASA), Houston, Texas for funding the research contract No. NAG 9-614. The authors would also like to thank the Center for Manufacturing Research and Technology Utilization of Tennessee Technological University and the Department of Electrical Engineering, Tennessee Technological University for supporting this research.

This report was extracted with minor revisions from the report for M.S. degree in Electrical Engineering submitted by Ajmal Khan to the Graduate School of Tennessee Technological University, comprising the research carried out under the supervision of Dr. P. K. Rajan.

TABLE OF CONTENTS

Chapter Page

Chapter Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

ABBREVIATIONS

AMF	-	Amplitude-mostly Filter
BPOF	-	Binary Phase-only Filter
DMD	-	Deformable Mirror Device
FT	-	Fourier Transform
GMACE	-	Gaussian Minimum Average Correlation Energy
LCLV	-	Liquid Crystal Light Valve
MACE	-	Minimum Average Correlation Energy
Mod-MACE	-	Modified Minimum Average Correlation Energy
MF	-	Matched Filter
MSF	-	Matched Spatial Filter
OFE	-	Optical Feature Extraction
OPOF	-	Optimal Phase-only Filter
OPR	-	Optical Pattern Recognition
PCE	-	Peak-to-Correlation Energy
PMF	-	Phase-mostly Filter
POF	-	Phase-only Filter
PSR	-	Peak-to-sidelobe ratio
RMACE	-	Real-valued Minimum Average Correlation Energy
SDF	-	Synthetic Discriminant Function
SLM	-	Spatial Light Modulator
SNR	-	Signal-to-noise Ratio

CHAPTER 1

Introduction

During the past two decades, there has been considerable growth of interest in problems of pattern recognition and image processing. Applications in this area include character recognition, remote sensing, image compression, image enhancement (for example, in medical diagnosis), speech recognition, archaeology, industrial automation (machine part recognition, automatic inspection), and target tracking systems [1]. These expanding applications and the advent of high speed digital computers with increased storage capabilities have further fueled research into the area of pattern recognition.

There are many digital electronic pattern recognition systems which often involve non-real-time operations. However, many military and space applications such as missile guidance, vehicle tracking, and automated lander guidance in aerospace missions involve certain real-time operations. Optical pattern recognition (OPR) offers an attractive solution for such applications because of its inherent parallel processing and high-speed operation capabilities [2]. Several signal processing operations such as Fourier transform (FT), convolution, correlation, and spectral analysis can be carried out more efficiently using optical techniques than with their

electronic counterparts [3]. Optical data processing therefore represents an attractive alternative for potential commercial applications such as robotics, automated product inspection, and other civilian fields. In the next section, a brief overview of the OPR techniques will be given.

1.1. Correlation-Based Optical Pattern Recognition

Two basic approaches to optical pattern recognition are optical feature extraction (OFE), and optical correlation. In case of OFE, certain geometrical properties of the input object (for example, its edges) are computed. The results are subsequently processed (usually by a digital computer) for classification, employing certain preselected discriminant functions. In the optical correlation approach a filter is synthesized from the image of the reference object to be identified. The input image to be tested for the presence of the desired object is correlated with the filter function. The maximum resultant cross-correlation value is compared to a preselected threshold value to determine if the desired object is present in the input image scene. The location of the correlation peak determines the position of the desired object in the input image. Basic optical correlators do not provide invariance to distortions in the input image, for example, in-plane rotations. Hence, to achieve general distortion invariance, advanced techniques such as synthetic discriminant functions and other methods are used [4].

The correlation operation may be interpreted as a measure of similarity between two functions (for example, two images), and a similarity measure is the basis for any recognition process. In 1963, Vander Lugt proposed a hologram-based technique for constructing a matched spatial filter (MSF). With this technique, it became possible to construct arbitrary complex matched filters to detect a signal in a noisy background by optical means. The MSF when used in a frequency plane optical correlator forms an optical pattern recognition system. The most common type of such a system is illustrated by Figure 1.1.

An input scene at plane P_1 denoted by $f_1(x,y)$ is illuminated by laser light to form a diffraction pattern. This pattern is focussed by a lens L_2 on plane P_2. The lens L_2 produces the spatial Fourier transform of $f_1(x,y)$, denoted by $F_1(u,v)$, i.e.

$$F_1(u,v) = \mathcal{F}\{f_1(x,y)\}. \tag{1.1}$$

The MSF denoted by H(u,v) is derived from a reference pattern $f_2(x,y)$ and is placed at the focal plane P_2. It is basically the conjugate of the Fourier transform of the reference pattern, i.e.

$$H(u,v) = F_2^*(u,v). \tag{1.2}$$

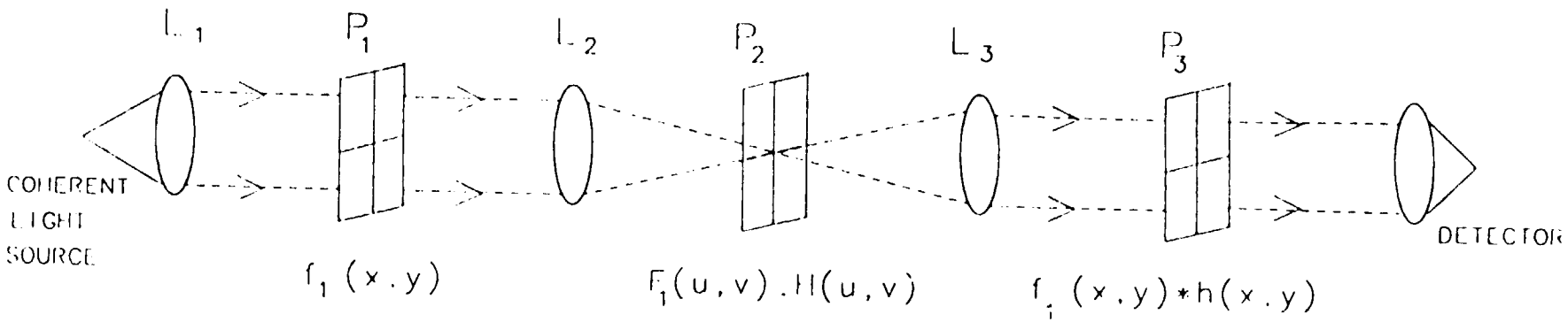

Figure 1.1. A Frequency Plane Optical Correlator

where

$$F_2(u,v) = \mathcal{F}\{f_2(x,y)\}, \tag{1.3}$$

and * denotes the conjugate operation. The incoming pattern of light from P_1 is diffracted by the MSF, thereby producing a second diffraction pattern. This resultant pattern on P_2 is the product of the transforms $F_1(u,v)$ *and* $H(u,v)$. Finally, the lens L_3 produces the inverse Fourier transform of this product, which is the correlation function. It can be written mathematically as

$$\mathcal{F}\{f_1(x,y) \otimes f_2(x,y)\} = F_1(u,v).H(u,v) \tag{1.4}$$

$$= F_1(u,v).F_2^*(u,v), \tag{1.5}$$

and

$$f_1(x,y) \otimes f_2(x,y) = \iint f_1(x',y')\, f_2^*(x'-x, y'-y)\, dx'\, dy', \tag{1.6}$$

where $\otimes$ denotes the correlation operation. If the input and reference patterns are identical, then the operation is called autocorrelation.

Thus

$$\mathcal{F}\{f_1(x,y) \otimes f_1(x,y)\} = |F_1(u,v)|^2. \qquad (1.7)$$

The correlation function of Eq. (1.6) is obtained at the output plane P_3. As mentioned earlier, the correlation peak value is used to determine if the input image $f_1(x,y)$ is indeed the desired image to be recognized.

Optical systems (such as the one described above) which use the Fourier transform property of a lens offer certain unique advantages over electronic systems. The inherent two-dimensional nature of an optical system permits it to accept large two-dimensional arrays of data from a camera, slide, or transparency very easily. A second advantage enjoyed by an optical system is its capability for parallel computation, that is, all the data points of an image pass through the optical system simultaneously, i.e., in parallel. A third advantage of an optical system is its ability to operate at tremendous data rates. Since the signal travelling through a passive optical system propagates at the speed of light, its throughput is enormous. Finally, a fourth advantage of an optical system is that the Fourier transform is relatively simple to perform. Thus with the FT as the basic building block, it is straightforward to design systems to perform correlation, convolution, matched filtering, etc. [5].

It is apparent, however, that there are certain tasks needed of a pattern recognition system that are difficult or impossible to perform by a purely optical system. One problem is that simple optical systems by themselves cannot be used to make decisions. The simplest type of decision task might be a comparison of the output of an optical system with a stored value, and this cannot be performed presently without help from electronics. Thus, purely optical systems can be designed to perform specific tasks (analogous to a "hard-wired" electronic computer), but are not flexible like a programmable general purpose digital electronic computer. The idea of combining digital electronic computer technology with optical systems thus follows quite naturally, as a means of applying the advantage of optics to a wider range of problems. These systems are called hybrid, since they are composed of both optical and electronic subsystems.

1.2. Motivation for the Study

Since the introduction of the frequency plane correlator by Vander Lugt, matched spatial filters (MSFs) have become popular in optical pattern recognition systems. These MSFs yield the highest possible signal-to-noise ratio (SNR) when detecting a known signal (image) corrupted by additive white noise [6]. However, MSFs do not produce sharp peaks and they are light-inefficient, that is, the ratio of the output light to input light power is only about 44 percent [7]. Further, MSFs are complex in nature; hence, their use in optical correlators requires the representation of both the magnitude

and phase of the filter function. Also, MSFs are sensitive to distortions in the input test images.

High speed interface devices known as spatial light modulators (SLMs) are being developed to replace the film transparencies that have been traditionally used as input images and as MSFs in conventional optical correlators. However, several SLMs of current interest can function only in a phase-mostly mode, and thus cannot accommodate the complex nature of the MSFs.

The concept of MSFs has been extended in recent years to several types of distortion-invariant filters. The minimum average correlation energy (MACE) [8] filter is one of them. This filter attempts to minimize the sidelobe levels in the entire correlation plane, while still allowing control of the user-specified correlation peak value at the origin. Thus the MACE filter produces sharp correlation peaks facilitating easy detection in the correlation plane. In the MACE filter design, training images are used which are sufficiently representative of all the expected distortions (that is, 3-D distortions). This is to reduce the filter sensitivity to the distortions. The MACE filter will be discussed in greater detail in Chapter 2.

As mentioned earlier, SLMs are used in optical correlators at the input and filter planes. The various SLMs, constructed on different media, have different constraints on what is physically achievable in that medium. Since the SLM medium is used to construct the filter in the Fourier domain

of an optical processor, these physical constraints will also be imposed on the filter in the Fourier domain. Since the physical characteristics of an SLM are not taken into account in the design of the above mentioned MACE filter, they specify arbitrary complex values, and hence they cannot be implemented on currently available SLMs.

In References [9] and [10], correlation filters have been suggested that incorporate the physical characteristics of a given SLM in their design. Juday [9] has developed a set of necessary conditions for optimizing an optical correlation filter constrained by an amplitude-phase coupled SLM. Farn and Goodman [10] have demonstrated a technique for an optimal filter given an arbitrary region of realizability, which maximizes the output correlation peak and they have presented a fast algorithm for the design. However, these SLM-constrained filters do not take into account possible distortions in the input images and therefore, there is a need for the development of distortion-invariant filters which are constrained to the region of realizability of the physical characteristics of a given SLM. In Reference [3], a real-valued MACE filter was developed for implementation using real-valued SLMs. However, until very recently, no composite filters had been developed, which could accommodate an arbitrarily constrained SLM.

1.3. Objectives

In order to overcome the difficulties mentioned above, the principal objective of the research leading to this report

was chosen as the development of a technique to incorporate the physical characteristics of a given SLM into a distortion-invariant correlation filter design. The type of SLM constraint considered was one in which there existed a cross coupling between the amplitude and phase. The MACE filter was chosen for optimization, since it minimizes the average energy over the correlation plane and maintains the user-specified peak value at the origin. This way the sidelobes in the output correlation plane are much smaller than the value at the origin. This facilitates easy detection of the correlation peak in the entire correlation plane. The simulated annealing algorithm was used to perform the minimization of the correlation energy function for this constrained MACE filter. The performance of this filter was to be evaluated on input images with and without background noise.

1.4. Outline of the Report

This report is organized as follows. In Chapter 2, various distortion-invariant pattern recognition filter designs from the literature are reviewed. Chapter 2 also discusses the characteristics of currently available spatial light modulators (SLMs), and some filter optimization techniques under the constraints of the physical characteristics of an SLM. In Chapter 3, the development of the SLM constrained distortion-invariant filter is discussed, along with the algorithm used for its synthesis. Chapter 4 presents simulation results used to evaluate the performance of this constrained composite filter, along with a discussion

of the results. In Chapter 5, a summary of results and conclusions derived from the study are presented along with recommendations for future work.

CHAPTER 2

Correlation-Based OPR Filters

As mentioned in Chapter 1, in the field of correlation-based optical pattern recognition systems, matched spatial filters are optimal in the sense of maximum signal to noise ratio (SNR). However, these filters perform poorly when the input test images are subjected to distortions like in-plane or out-of-plane rotations. In this chapter, a review of some correlation-based distortion-invariant pattern recognition filters proposed in the literature is presented. In particular the MACE filter design is discussed in detail.

Recently, spatial light modulators (SLMs) are being used to implement filters in the frequency plane of optical correlators. Most currently available SLMs cannot encode fully the complex-valued frequency response of the optical pattern recognition filters that have been proposed by researchers. Some filter design techniques proposed in the literature which take into account the constraints imposed by the SLMs will be reviewed in latter sections of this chapter.

2.1. Distortion-Invariant Pattern Recognition Filters

Various approaches have been advanced in recent years to achieve general distortion invariance in optical pattern recognition. Casasent [11] detailed a generalized method which

uses a synthetic discriminant function (SDF) to form a distortion-invariant MSF for use in optical correlators. The SDF is a composite function of all the training images which are deliberately distorted versions of the reference image. Thus the SDF filter is a linear combination of the training images, and it is designed to yield user-specified cross-correlation values at the origin for all the training images. The expectation is that this filter will correlate equally well not only with the training images, but also with other distorted versions of the reference image which are within the distortion range of the training set images. A disadvantage with the SDF filter is that while the specified correlation value is obtained at the origin, there is a possibility of large sidelobes in the correlation plane. Thus, the next step was to design composite filters which will minimize the sidelobe levels in the correlation plane.

Mahalanobis et al.[8] proposed the minimum average correlation energy (MACE) filter which results in a sharp correlation peak of a specified value at the origin due to the minimization of the average correlation plane energy. As in the SDF filter, training images are used to reduce the sensitivity of the filter to distortions. In the following, the MACE filter design is discussed.

Throughout this report, the images as well as the filters are assumed to be in the discrete domain. The images and filters consist of arrays of dimension NxN obtained by discrete sampling of continuous signals. However, for

convenience, these arrays are represented by vectors of size dx1 (d = NxN) through lexicographical ordering of their rows. Boldface letters are used to denote vectors and matrices, and lower case letters denote scalar quantities. Image and filter sequences are identified by an overbar notation. Uppercase symbols refer to the frequency plane terms, while lower case symbols represent quantities in the space domain.

Let the data sequences $\bar{x}_i(n)$, $i = 1, 2, \ldots, N_t$, represent the training image sequences, where N_t is the number of images in the training set, and $\bar{X}_i(k)$ denotes the DFT sequence of $\bar{x}_i(n)$, represented as a column vector $\boldsymbol{X}_i$ of dimension d. The matrix $\boldsymbol{X}$ with column vectors $\boldsymbol{X}_i$ is denoted by

$$\boldsymbol{X} = [\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_{N_t}]. \qquad (2.10)$$

The vector $\boldsymbol{h}$ represents the filter sequence $\bar{h}(n)$ in the space domain, and the vector $\boldsymbol{H}$ its DFT $\bar{H}(k)$ in the frequency domain. The average correlation plane energy for all the N_t images is given by

$$E_{ave} = \frac{1}{N_t} \sum_{i=1}^{N_t} \left[\frac{1}{d} \sum_{k=1}^{d} \left(|\bar{H}(k)|^2 \, |\bar{X}_i(k)|^2 \right) \right]. \qquad (2.11)$$

Now, Eq. (2.11) can be written in vector-matrix notation as

$$E_{ave} = \boldsymbol{H}^{+}\boldsymbol{D}\boldsymbol{H}, \tag{2.12}$$

where the superscript + denotes the conjugate transpose operation. $\boldsymbol{D}$ is a diagonal matrix of size dxd given by

$$\boldsymbol{D}(k,k) = \frac{1}{N_t}\sum_{i=1}^{N_t}\frac{|\overline{X}_i(k)|^2}{d}. \tag{2.13}$$

The MACE filter is designed so as to minimize the average correlation energy given by Eq. (2.12), while at the same time the value of the correlation function at the origin equals a user-specified value. The constraint at the origin of the correlation plane can be written as

$$g_i(0) = \frac{1}{d}\sum_{k=1}^{d}\overline{X}_i(k)\,\overline{H}(k) = u_i, \tag{2.14}$$

where u_i is the user-specified constraint value at the origin. For all the training set images Eq. (2.14) can be written as

$$\boldsymbol{X}^{+}\boldsymbol{H} = \boldsymbol{u}, \tag{2.15}$$

where $\boldsymbol{u}$ is the $N_t x1$ output constraint vector. Thus E_{ave} in Eq. (2.12) is minimized subject to the constraints in Eq. (2.15) using the Lagrange's Multiplier method.

This leads to the following filter [8]

$$H_{MACE} = D^{-1} X (X^{+} D^{-1} X)^{-1} u. \qquad (2.16)$$

Although MACE filters produce sharp correlation peaks, they appear to have two drawbacks. First of all, there is no noise tolerance built into these filters. Secondly, these filters seem to be more sensitive to non-training images than other composite filters [12]. Some modifications to the MACE filter have been proposed to overcome these problems.

To make the composite filters more robust against the non-training images, Casasent and Ravichandran [13] proposed the design of the Gaussian MACE (GMACE) filters where the output correlation was constrained to be a broad Gaussian, rather than a sharp function. Sudarshanan et al.[14] suggested a compromise composite filter wherein the MACE filter design was modified to accommodate the requirements of both a good noise performance and a sharp correlation peak output.

2.2. Filter Implementation in Optical Correlators

So far in this chapter, some of the composite filters designed for distortion-invariant pattern recognition, i.e., filters which are insensitive to distortions, have been discussed. Another important consideration in the design of optical correlation filters is their practical implementation using available electro-optical devices. As mentioned in Chapter 1, in 1963, Vander Lugt [15] demonstrated a simple hologram-based optical technique for synthesizing frequency-

plane masks for coherent processors. This technique recorded a complex filter function on a photographic film with the help of an interferometric system [15]. In such a system, coherent light having an amplitude transmittance equivalent to the desired impulse response of the filter is made incident on a lens. The resultant Fourier transform of the impulse response is incident on the photographic film which records the complex filter function. Lohmann and Paris [16] proposed an easier method for constructing complex filters. They suggested using a computer-aided plotter to draw the holograms which would be used to realize the filters. The holograms would then be recorded on a film. This eliminated the rather difficult task of optically synthesizing the filter on the photographic film as suggested by Vander Lugt.

Since the introduction of these filter implementation techniques, various design algorithms for OPR filters were proposed by researchers. However, the above discussed techniques for filter implementation were found to be unsuitable for certain real-time applications of OPR systems. Such applications need the capability wherein the input images as well as the reference filters can be changed rapidly. Thus, in order to take advantage of the phenomenal processing speed of an optical correlator system, high-speed interface devices, known as Spatial Light Modulators, were introduced to replace the holograms.

2.2.1. Spatial Light Modulators (SLMs)

SLMs are the basic building blocks of OPR systems which are used to implement the input images and filters. They encode the phase and amplitude information of an image or a filter onto an incident beam of coherent light as a function of space. There are two types of SLMs, one which can be addressed electrically and the other that can be optically addressed. In the case of the former type, voltage or charge is the control signal (for example, the Deformable Mirror Device (DMD)), while light intensity is the control signal for the latter type (for example, the Hughe's Liquid Crystal Light Valve (LCLV)) [17].

However, there is a limitation on the currently available SLMs with regard to the actual implementation of a filter on them. Ideally, an SLM should be able to adjust both phase and amplitude as required by a recognition filter. For all currently available SLMs, one control signal (either optical or electrical) is used to achieve the phase and amplitude modulation. Hence, the phase and amplitude cannot be controlled independently. In a DMD, the control signal not only modulates phase but also changes the amplitude from unity, though to a lesser extent, and vice-versa for the LCLV. Hence, these are appropriately referred to as phase-mostly filters (PMF) and amplitude-mostly filters (AMF),

respectively. This fact will become clear for the case of the DMD, when its practical operation is discussed in the next section.

2.2.2. The Deformable Mirror Device (DMD) SLM

Figure 2.1, reproduced from [18], shows a cutaway view of a section of the deformable mirror device SLM. The DMD is composed of an array of 128x128 pixel elements with each element consisting of four reflective rectangular flaps hinged on the corners. These are known as cantilever-beam deformable mirrors. The flaps are drawn downward by an electrostatic charge maintained on the electrode under the surface of each pixel element [18]. It is this mirror deflection that is used to modulate light [19]. Since each of the flaps is about 0.5 mil square, a deflection of about one degree will cause the inner tip of the flap to displace through π radians of phase using laser light (Wavelength = 632.8 nm) [19]. Thus, the DMD is primarily a spatial phase-modulator that permits the phase at a given pixel element to be controlled over a continuous phase range through an electrical control signal. Since the pixel flaps move through a range of angles to the DMD surface, there is some additional amplitude modulation of the light due to the signal deflection.

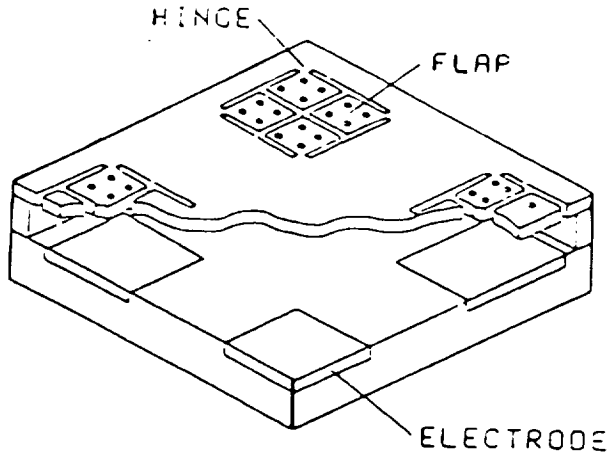

Figure 2.1. Cutaway View of a Section of the Deformable Mirror Device (DMD) SLM [18]

SLMs form an integral part of optical correlator systems today. For an optical correlator to be generally useful, the input plane SLM must operate in real-time. In fact for most applications, the filter plane SLM must operate at a much faster rate. This is especially true if a large number of filters need to be changed at a rapid rate, before a correlation is detected. The DMD is faster than most SLMs, and exceeds the speeds of common liquid crystal displays [20]. Hence, the DMD is a suitable SLM for use in an optical pattern recognition system. In the next subsection, some filter designs which accommodate the modulating characteristics of available SLMs are discussed briefly.

2.2.3. Correlation-Based Filters for SLMs

Since the time Vander Lugt introduced the matched spatial filter, various filter designs were proposed by researchers. Some of these were discussed in Section 2.1 of this chapter. All these methods presuppose the use of continuous amplitude-phase filters in the realized optical correlator. However, as discussed in the last two subsections, programmable, continuous amplitude-phase SLMs do not exist to date. Hence, the utility of these filter designs in practical correlators is severely limited. On the other hand, programmable SLMs limited to quantized levels of amplitude or phase are presently available. In anticipation of the availability of these devices, models of the SLMs like the DMD, and filters that could be synthesized on these SLMs were developed by

researchers. The modifications of the classical MSF to get the phase-only filter (POF) [21] and the binary phase-only filter (BPOF) are among the most important, since they are easy to implement on available SLMs such as the Magneto-Optic SLM.

The POF is identical to the MSF, except that its magnitude is set to unity for all frequencies. In POFs, since the amplitude transmittances are set to be constant, the continuous phase is used to synthesize the filters on existing phase-modulating SLMs. BPOFs are a simplified version of the POFs. Here, the continuous phases of the POF are quantized into two phase levels. The pixels in a BPOF take a value of +1 or -1 depending upon their respective phase angles. Generally the POF and BPOF have to be implemented on a phase-modulating SLM. Several researchers have reported methods for optimizing the conventional POFs and BPOFs based on various performance measures. Some of these are discussed in the next subsection.

2.2.4. Filter Optimization

Techniques

Vijayakumar and Bahri [6] investigated theoretically the optimality of the conventional POF. They proved that with proper selection of the filter passband the conventional POF yields the highest signal to noise ratio (SNR) among all filters. The resulting filter (with the corresponding filter passband) is termed the Optimal POF (OPOF). This concept of optimizing an optical correlation filter constrained to a particular SLM has proven to be an important subject for research with the anticipated availability of SLMs with

different operating characteristics. Accordingly, much work has been devoted to this subject recently. Downie and Reid [22] studied the optimization of BPOFs to obtain optimal peak-correlation response and peak to sidelobe ratio. Recently, Downie [23] investigated the design of BPOFs that maximize some other filter performance criteria like SNR and their ability to discriminate desired objects from the undesired ones.

The design and simulations of the phase-encoded and binary phase-encoded filters and their optimization by the various researchers were based on the phase-only property of the SLMs. They did not take into account the "phase-mostly" nature of these devices, which actually modulate the amplitude in addition to the phase as observed in Section 2.2.1 of this chapter. Thus, some other filter optimization techniques were proposed which did precisely that. But, as seen above, the designs of an optimal filter were considered only in special cases (e.g., BPOF design) [6,22-23]. Therefore, there is a need to design an optimal filter, given a general realizability constraint for an SLM.

2.2.5. Filter Optimization for Cross-Coupled SLMs

Ideally, an SLM should adjust both phase and amplitude independently. However, as observed for the DMD in Section 2.2.2 the single control parameter, namely, the voltage, modulates the phase and amplitude simultaneously. This is known as the cross coupling between phase and amplitude. The

actual cross coupling between the phase and amplitude may be adjusted during the manufacture of the device. But, given that the SLM is controlled through a single parameter (electrical signal in the case of the DMD), one cannot achieve any arbitrary combination of phase and amplitude. Rather, the relationship between the phase and amplitude is fixed for all the pixel elements, for a given SLM. The challenge is to obtain an optimal filter within the constraints of the one-parameter combination of phase and amplitude for each pixel value of the filter.

Juday et al.[18] showed a simulation of optimizing an amplitude-phase cross-coupled filter by a brute force relaxation method of Juday and Daiuto [24]. They considered the variation of the SLM phase to be quadratic in logarithm of the amplitude. Figure 2.2, reproduced from [18], shows this relationship between the phase and amplitude. As the electrical signal S varies, the response of the DMD is constrained to move along the curve. To start with, a POF is computed and applied to the filter DMD. The amplitude and phase values for the (m,n)th element of the conventional matched filter (MF) is represented by the point MF(m,n). A POF device response would correspond to the point POF(m,n). However, the needed filter pixel values are constrained to lie only on the curve. Retaining the phase value of the POF for this (m,n)th pixel under the operating constraint, the actual response to the calculated filter occurs at position $CMF_0(m,n)$ (in Figure 2.2).

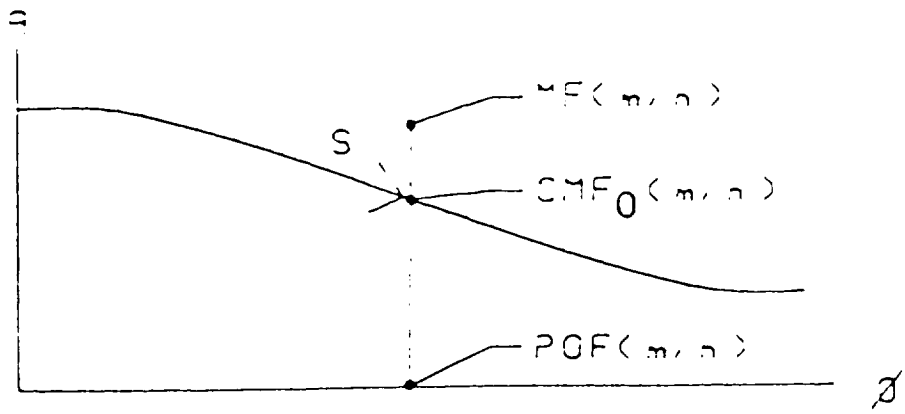

Figure 2.2. The Amplitude-Phase Relationship used to Synthesize the Cross-Coupled Filter [13]

The filter optimization is continued by varying the values of the filter DMD, pixel by pixel, until a maximum correlation is observed. After all the pixel values have been adjusted, the correlation function value of the filter is compared to that of the last filter. The procedure is terminated when this difference is less than some predetermined value. Thus, the starting filter is CMF_0, and it is relaxed to the optimal final filter CMF_f. The details of this algorithm are given in Reference [24]. The computation needed for this method is considerable. Juday [9] has proposed another analytical technique to solve this same filter optimization using the method of calculus of variations. This new method considerably improves on the relaxation algorithm discussed above, by reducing the amount of computation needed.

The filter synthesized using the above technique is optimal in the sense of obtaining the maximal correlation peak. However, it does not minimize the correlation plane energy to yield low sidelobe levels as in the case of the MACE filter. One of the objectives of the research presented in this report was to do precisely that.

2.2.6. Correlation Filters for Arbitrarily Constrained Devices

Farn and Goodman [10] have proposed a theory for the optimal design of a correlation filter, given any arbitrary realizability constraint. The filter is optimal in the sense of maximizing the intensity at the output correlation peak.

The constraint depends upon the physical modulating characteristics of a given SLM. They have also proposed a fast algorithm to design the filter. The SLM constraints for the filter could be phase-only modulation, binary phase-only modulation, or amplitude-phase cross coupling.

This technique and the one discussed in the previous section represent the first attempts at optimizing a filter, given any general constraints for the operation of an SLM. However, all the optimization techniques for constrained filters discussed so far are intended for the recognition of a single image. There have been some attempts aimed at designing distortion-invariant correlation filters which are suitable for implementation on available SLMs and they are discussed in the following section.

2.2.7. Distortion-Invariant Correlation Filters for SLMs

As mentioned before, all the composite filter designs discussed at the beginning of this chapter have assumed the availability of programmable, continuous complex valued SLMs. In reality, this is not true. Hence, the focus of research has shifted to developing distortion-invariant filters for optical correlators that are capable of being implemented on SLMs with limited modulation capabilities.

2.2.7.1. SDF-POFs and SDF-BPOFs. Horner and Gianino [25] applied the phase-only concept to a conventional SDF. Basically, they computed an SDF similar to the one proposed by

Casasent [11], and derived its POF version by setting the magnitude to unity for all frequencies. This is known as the SDF-POF. They also developed a simplified version of the SDF-POF using biphase quantization. Here, the phase values of the SDF-POF are quantized to just two values depending upon their original phase values. This approach is similar to that of designing a BPOF, and is referred to as an SDF-BPOF [25]. However, design of the SDF-POFs, and BPOFs using this technique is incorrect because the filter modulation operator (for example, POF or BPOF) is applied to a conventional SDF filter after it is synthesized. This will change the information content contributed to the filter from each training image. Hence the correlation of such a filter with each training image will not give the specified peak at the origin.

Kallman [26] has suggested a method of directly constructing distortion-invariant phase-only filters by optimizing the peak-to-sidelobe ratio for a set of training images. Although this method yields filters with desirable qualities, the technique is computationally intensive, and yields little information about the theory of distortion invariant filter construction [27]. Also, it does not provide a specified correlation peak for each training image.

2.2.7.2. Composite filters constrained to SLM modulating characteristics. Jared and Ennis [27] have considered a modification to the conventional SDF by including the filter modulation (due to the SLM) in the synthesis process. They

have considered POFs and BPOFs, but the technique is general enough to be applied to filters of any modulation. In this method, the actual filter that will be used to produce the specified correlation peak response is explicitly used during the SDF synthesis [27].

Balendra and Rajan [28] have developed a real-valued MACE (RMACE) filter for implementation on available real-valued SLMs. It is a modification to the MACE filter which cannot be used with the existing SLMs due to its complex frequency response. Similar to the MACE, the RMACE is designed to minimize the average correlation energy under the constraints of the user specified correlation peak outputs.

As seen in this section, a few distortion invariant correlation filters which take into account the modulating constraints of existing SLMs have been proposed by researchers. However, the SLM constraints considered have been special cases (e.g. Phase only, Binary Phase only, and real). Thus, there is a need to develop distortion-invariant composite filters which can be implemented on any arbitrarily constrained SLM, for example, an amplitude-phase coupled SLM.

In Reference [29] Kumar has proposed a technique which uses a relaxation algorithm to design a SDF filter for implementation on arbitrarily constrained devices. The criterion used for optimization in that case was such that the Peak-to-Correlation Energy (PCE) [30] for the desired class of objects is maximized. The approach used in this report research for the design of the constrained composite filter is

different from the above in the optimization method used. Also, here the average correlation plane energy was sought to be minimized while maintaining the user-specified values at the origin of the output correlation plane.

2.3. Summary

This chapter focussed attention on two important areas of filter design techniques for optical pattern recognition. One of them is the development of correlation-based distortion-invariant filters, and the other is the implementation of these filters on currently available spatial light modulators. In this report, the development of a composite filter for implementation on an arbitrarily constrained SLM is studied. In the following chapter, the optimization algorithm used to synthesize this filter is discussed in detail.

CHAPTER 3

SLM-Constrained MACE Filter

A number of filter design techniques that have been proposed to achieve distortion-invariant pattern recognition were reviewed in Chapter 2. The minimum average correlation energy (MACE) filter is one such filter which produces a sharp correlation peak in the output facilitating easy detection. The MACE filter produces higher peak-to-sidelobe ratios than the SDF filters. Hence, in this report the design of MACE filters is considered. The need for an SLM-constrained MACE filter is first discussed and then an algorithm for its synthesis is presented.

3.1. The Need for an SLM-Constrained MACE Filter

As mentioned in Chapter 2, SLMs are used in OPR systems to implement the filters. Whereas a designed filter may have arbitrary complex values, the SLM, as noted earlier, can accommodate only a constrained set of values. If the correlation-based OPR filters are to be implemented on such SLMs, then the modulation characteristics of SLMs must be included in their design.

One such SLM, namely, the DMD, has good potential for use in OPR systems due to its advantages over other SLMs as discussed in the previous chapter. Hence, there is a need to

develop filters which can be implemented on the DMD. Most of the SLM-constrained filter designs proposed in the literature do not take into account distortions in the input images. Some of these designs were discussed in Sections 2.2.4-2.2.7 of Chapter 2.

As discussed in Chapter 2, the MACE filter is attractive for use in Optical Correlators since it minimizes the average correlation energy over the output correlation plane, while at the same time maintaining the user-specified sharp correlation peak. It is proposed in this report to apply the MACE design concept to develop a method to design a SLM-constrained distortion-invariant filter. This SLM-constrained MACE filter can be implemented on a DMD with its phase-magnitude constraints.

3.2. Development of the SLM-Constrained MACE Filter

The problem at hand can be stated as follows. It is required to design a composite filter which will incorporate the possible distortions (for example, in-plane rotations) in the input test images, and at the same time the filter pixels are constrained to take on values which obey the physical characteristics of a given SLM. The SLM-constrained MACE filter is such that the correlation function levels are minimized at all the points in the output plane except at the origin where the response must have a user-specified value. Of course the operating characteristics of the SLM should be available beforehand.

The required composite filter is synthesized from a set of training images which represent the distorted versions of the reference image. The SLM-constrained MACE filter, like the original MACE filter, is synthesized in the discrete frequency domain. The column vectors $\boldsymbol{X}_i$ and $\boldsymbol{H}$ denote the i^{th} training image and filter, respectively, in the frequency domain. The matrix $\boldsymbol{X}$ is formed by arranging all the training image vectors $\boldsymbol{X}_i$, $i=1,2,\ldots,N_t$, as columns of that matrix, where N_t denotes the number of images in the training set. The training image vectors $\boldsymbol{X}_i$ and filter vector $\boldsymbol{H}$ are of dimension d (equal to the number of pixels in the reference image, namely, NxN). Minimization of the correlation function values at all the points in the output plane can be achieved by minimizing the average correlation plane energy, which is given by

$$E_{ave} = \boldsymbol{H}^{+}\boldsymbol{D}\boldsymbol{H}, \tag{3.1}$$

where + denotes the conjugate transpose operation. The diagonal matrix $\boldsymbol{D}$ is given by

$$\boldsymbol{D} = \frac{1}{N_t}\sum_{i=1}^{N_t} D_i, \tag{3.2}$$

where

$$D_i(k,k) = \frac{|\boldsymbol{X}_i(k)|^2}{d}. \tag{3.3}$$

While minimizing Eq. (3.1), the correlation function value at the origin for each of the training images is to be maintained at a user specified value. This will facilitate easy detection of the peak value at the origin in the output plane. The correlation peak constraint for the i^{th} training image is given by

$$g_i(0) = \frac{1}{d} \boldsymbol{X}_i^* \boldsymbol{H} = u_i, \tag{3.4}$$

where u_i is the user-specified peak value for the i^{th} training image.

Next, the constraints imposed by the modulating properties of a given SLM need to be incorporated in the design of the filter. In a single-parameter SLM like the DMD, the phase and magnitude at a given pixel element are controlled by the excitation voltage applied at that pixel element. In such a case, there is no independent control over the phase and magnitude of the filter pixels. This is known as the phase-magnitude cross coupling for a given SLM, and is dependent on the manufacturing process of the device. The experimental data corresponding to the physical characteristics of the DMD is available in the form of a table with two columns. The entries in the first column of the table correspond to the phase values which represent the phase modulating capabilities of the SLM. The entries in the second column give the respective magnitude values for each phase

value. This table limits the complex pixel values that the required filter can acquire so that it can be implemented on a given SLM.

Thus, the filter design problem can be viewed as a constrained optimization problem and can be summarized as follows. The aim is to determine an optimal filter by selecting its pixel values from among the phase-magnitude value pairs given by a table such that the correlation plane energy given by Eqn. (3.1) is minimized subject to the peak-value constraints of Eqn. (3.4). Thus Eqn. (3.1) is the objective function here. The latter part of the problem, namely, minimizing Eqn. (3.1) subject to the constraints of Eqn. (3.4) can be modified into a new single objective function. This can be done by including in the new objective function some penalty for violating the constraints.

The correlation peak constraints of Eqn. (3.4) can be written as

$$\begin{aligned} Re\{\boldsymbol{X}_i^* \boldsymbol{H}\} &= u_i \\ Im\{\boldsymbol{X}_i^* \boldsymbol{H}\} &= 0. \end{aligned} \tag{3.5}$$

Using Eqns. (3.1) and (3.5), the new objective function can be written as

$$E(\boldsymbol{H}) = \boldsymbol{H}^+ \boldsymbol{D} \boldsymbol{H} + \sum_{i=1}^{N_t} K_{1i} [\, Re\{\boldsymbol{X}_i^* \boldsymbol{H}\} - u_i \,]^2 + K_2 \sum_{i=1}^{N_t} [\, Im\{\boldsymbol{X}_i^* \boldsymbol{H}\} \,]^2. \tag{3.6}$$

Here, the constraints of Eqn. (3.5) are used to introduce quadratic penalty functions into the original objective function.

In Eq. (3.6) the K_{1i}'s and K_2 are some positive constants greater than 1. The values for these constants will depend upon the specified constraint values u_i. For example, in a two-class pattern recognition problem, the u_i values for the training images of the two classes will be different. Therefore the weightage for the K_{1i}'s will be different for the different u_i's. The ratio between the K_{1i}'s should be the same as the ratio between the corresponding u_i's. Thus, the problem of minimizing Eqn. (3.1) subject to the constraints of Eqn. (3.5) is reformulated into a problem of minimizing a single objective function with the constraints of Eq. (3.5) included within that objective function. Now the required optimal filter $\boldsymbol{H}$ can be determined such that Eqn. (3.6) is minimized under the constraints of the SLM modulating characteristics. The optimization is commenced with small values for the penalty constants and is continued with gradually increasing values such that the constraints are not violated and the objective function converges to a minimum. The filter thus obtained is the SLM-constrained MACE filter. The system variables in this optimization are the filter pixel values.

It is expected that the correlation function of this filter with the training and non-training set images evaluated at the origin will give values which are close to the user-specified values at the origin. Of course, the non-training images should be within the distortion range of the training set images. The sidelobes in the correlation plane are also expected to be minimized so that the values at the origin will correspond to a peak and can be easily detected.

3.2.1. Optimization

Techniques

Several optimization techniques are available in the literature and the choice of a particular algorithm would depend upon the nature of the problem to be solved. Optimization problems can be broadly divided into two categories: those with continuous variables, and those with discrete variables. The optimization problem at hand as discussed in Section 3.2 basically falls into the second category. However, the SLM constraints given by the phase-magnitude data could be approximated by a mathematical equation. The constraint equation so formed could be included in the objective function of Eqn. (3.6) using the penalty function technique. The entire problem could then be viewed as an unconstrained minimization problem and it could solved using the algorithms from the first category some of which will be discussed here.

Methods of finding the minimum of a function of continuous variables generally involve the evaluation of the

function values, their first partial derivatives, and sometimes the second partial derivatives as well. Optimization techniques which make use of partial derivatives are known as gradient methods. The method of steepest descent is one such gradient technique. This method starts by choosing arbitrary values for the function variables. New values for the variables are determined by evaluating the first derivative of the function at their present values. This procedure is continued until the derivative becomes zero or close to it. A local minimum of the function is said to be reached at those values of the variables.

The conjugate-gradient method of minimization is an advanced form of the steepest descent method. The Fletcher-Powell [31] version of the variable-metric algorithm is an improved form of the conjugate gradient technique to minimize a quadratic function with no more than n steps (where n is the number of variables in the function). There is no scope for discussion of the details of this algorithm here. Basically it is a rapidly converging descent method.

It is important to note here that all these methods require calculation of the first derivative of the function. Hence they are not suitable for solving the optimization problem discussed in Section 3.2. First of all, analytical calculation of the first derivative of the function is impossible here; hence, a numerical technique would have to be used. Now, the number of variables involved here are very large. For a 128x128 size image, the total number of variables

are 16384. Using a numerical technique to determine the first derivative would be very time-consuming. Also, the gradient techniques do not perform very well when there are boundary conditions specified for the function variables. This is the case for the problem at hand because the filter pixel values have to be within the given tabular data.

Now the second category of minimization techniques, namely, those involving discrete variables, seem to be more applicable to the problem at hand. Several direct-search methods for multi-dimensional minimization are suggested in the literature. These methods attempt to determine a global minimum of the objective function by evaluating it at several points rather than by calculating first derivatives. One direct-search technique which has been found to be particularly effective in certain applications is that of Hookes and Jeeves [32], known as the Pattern Search technique. However, its effectiveness has not been evaluated for the case of a very large number of variables.

There is another category of optimization problems known as combinatorial optimization. These could be generally classified as belonging to the second category, namely, that of discrete variables. Combinatorial optimization involves problems of finding the optimum of an objective function defined on a limited solution space. With this definition, the problem discussed in Section 3.2 does seem to have some similarities to a combinatorial optimization problem. This is because the experimental data corresponding to the SLM

characteristics does constitute the limited solution space available to solve the minimization problem of Eqn. (3.6). An optimization technique known as simulated annealing (SA) was proposed by Kirkpatrick et al.[33] to solve combinatorial optimization problems. It is discussed in detail in the next section along with its applicability to the problem at hand.

3.2.2. Solving the Optimization Problem by Simulated Annealing

The simulated annealing technique is a comparatively new method and has been used in the recent past for solving complex optimization problems. It is a form of stochastic optimization, and is reported to perform well for problems involving a large number of variables in applications such as the design of computers [33], image processing [34], and encoding of a binary phase filter [35]. It is easy to adapt the SA technique to the problem at hand, namely one in which there is a limited solution space. This refers to the magnitude-phase constraints of the SLM. Also, the number of variables involved in this problem, i.e., the filter pixels, is very large. These factors lead to choosing the SA technique to solve the SLM-constrained MACE filter optimization problem.

The idea of simulated annealing originated from the metallurgical annealing process in a physical system. The metallurgical annealing process is a thermal process for obtaining low energy states of a system. The aim here is to determine the state of global minimum energy of the system. This is what the physicists call the ground state. The ground

state is reached by a slow cooling of the system from a high temperature to a low temperature. The cooling schedule must be controlled appropriately and must be slow enough so that the system does not get trapped in thermodynamically metastable states. These are the local minima of the system. Conversely, if the system is cooled too slowly then it will take an unnecessarily long time to reach the ground state.

The lower energy states of the system can be achieved by using an algorithm proposed by Metropolis et al.[36]. This procedure uses computer simulation for computing the equilibrium distribution of a system composed of a set of particles in a heat bath. At each step of this algorithm, a particle is given a small random displacement and the resulting change in the system energy ΔE is computed. The new energy state of the system is accepted if $\Delta E \leq 0$, i.e., if the displacement brings the system to a state of lower energy. If $\Delta E > 0$, the new state is accepted with probability

$$P(\Delta E) = \frac{1}{1 + \exp[\frac{\Delta E}{kT}]}, \tag{3.7}$$

where k is the Boltzmann constant and T is the temperature of the heat bath. So a move to a higher energy state is accepted in a limited way. The choice of $P(\Delta E)$ has the consequence that the system approaches the Boltzmann distribution after several iterations. This distribution properly describes the state of thermodynamic equilibrium for a given T. This basic

step is repeated many times until the system reaches its ground state. Kirkpatrick et al.[33] linked the annealing of physical systems with combinatorial optimization and called it simulated annealing.

When the SA technique is used to find the global minimum of an objective function, the probability of being trapped in local minima can be made very small. This is possible because in addition to accepting transitions leading to a decrease in function value, increases in the function value are also accepted in a limited way. This makes it possible to climb out of a local minimum of the function and approach the global minimum.

The objective function that is sought to be minimized using the SA technique is given by Eqn. (3.6). The system variables to be determined in this optimization problem are the pixel values for the filter $\boldsymbol{H}$. The number of variables is given by d=NxN, which is the size of the reference image, hence, of the filter too.

Thus the minimization of an objective function using the SA technique can be achieved by reducing a temperature parameter from an initially high value to a low value, and perturbing the system variables as the temperature changes. Let the energy to be minimized (given by Eqn. (3.6)) be $E(v_i)$, $i = 1, 2, \ldots, d$, where the v_i's are the function variables to be determined.

The optimization is initiated by starting with a random

configuration of v_i, and calculating the energy $E(v_i)$. Here, the v_i's are the phase values of the filter pixels and are selected from among the phase-magnitude data available in a tabular form. As mentioned in Section 3.2 of this chapter, this data represents the physical constraints of the SLM. It should be noted here that when a phase value for a filter pixel is picked from the table its corresponding magnitude value is also chosen to accommodate the phase-magnitude coupling characteristics of the SLM. The phase-magnitude pairs represent the complex values of the filter pixels.

Now, for every iteration the following steps are carried out. Each of the system variables v_i , $i = 1, 2, \ldots, d$, is randomly perturbed. The energy change ΔE caused by the random perturbation Δv_i for the i^{th} variable is calculated as follows

$$\Delta E = E_{new} - E_{old}, \tag{3.8}$$

where

$$E_{new} = E(v_i^{old} + \Delta v_i), \tag{3.9}$$

and

$$E_{old} = E(v_i^{old}). \tag{3.10}$$

The perturbation Δv_i may be positive or negative and its actual value will depend on the data from among which the

filter pixels can accept values.

Now, the phases of the filter pixels are treated as the system variables. When the phase of a filter pixel is perturbed, the given table is searched for the new phase. The corresponding phase-magnitude pair from the table forms the complex value for that pixel. If $\Delta E > 0$, then the perturbation is accepted, based on the acceptance probability given by

$$P(\Delta E) = \frac{1}{1 + \exp(\frac{\Delta E}{T})}, \qquad (3.11)$$

where T is the temperature parameter which includes the constant k also. To do this, random numbers uniformly distributed in the interval (0,1) are generated. The random number is compared with $P(\Delta E)$. If it is less than $P(\Delta E)$, the perturbed configuration of v_i is accepted, otherwise the previous configuration is retained. This is repeated for all the d variables. For the perturbed configuration of v_i, the energy difference ΔE is computed rather than evaluating the entire objective function of Eqn. (3.6). This method of calculating and updating the system energy is computationally efficient. The probabilistic technique of accepting higher energy values avoids the possibility of the system being trapped in local minima.

The temperature control parameter T should be decreased gradually at appropriate steps. At each temperature, the optimization should continue long enough so that the system

reaches a steady state. The method of decreasing the temperature, and determining the number of variable perturbations attempted at every temperature is known as the annealing schedule, and is problem dependent. For the SLM-MACE filter design problem the initial temperature was chosen to be one-half the initial value of the objective function. The temperature T was lowered exponentially as $T_n = (0.95)^n T_0$, where T_n is the temperature at the n^{th} stage and T_0 is the initial temperature. At each temperature, enough perturbations are attempted such that there are ten accepted perturbations for every system variable on the average (this amounts to 10d accepted perturbations), or the number of attempts exceeds 100 times the number of variables before ten perturbations per variable have been accepted. If the desired number of accepted perturbations is not achieved at three successive temperatures, the system is considered to have reached its lowest energy state, and the optimization is concluded. This is the terminating criterion for the algorithm.

Thus, simulated annealing is used for minimization by first 'melting' the system being optimized at a high effective temperature (the temperature used as a control parameter), and then gradually lowering the temperature at appropriate stages until the system 'freezes' and no further changes occur.

3.2.3. Algorithm

An algorithm to construct the SLM-constrained MACE filter is presented below.

1. Obtain the training images $\overline{x_i}$, $i = 1, 2, \ldots, N_t$, by applying the various levels of distortions such as in-plane rotations, scale changes, etc., to the reference image(s), where N_t is the total number of images in the training set.

2. Compute the two-dimensional Fourier transform of these training set images as $\overline{X}_i$, $i = 1, 2, \ldots N_t$.

3. Order the elements of $\overline{X}_i$ lexicographically to obtain the column vectors $\boldsymbol{X}_i$, $i = 1, 2, \ldots, N_t$ of size dx1 where, d = NxN.

4. Determine the diagonal matrix $\boldsymbol{D}$, where its diagonal elements are given by

$$\boldsymbol{D}(m,m) = \frac{1}{N_t . d} \sum_{i=1}^{N_t} |\boldsymbol{X}_i(m)|^2 \qquad (3.12)$$

5. The next part of the algorithm involves determining the filter $\underline{\mathbf{H}}$, such that the following objective function is minimized, using the simulated annealing algorithm, as described in the previous section.

$$E = \boldsymbol{H}^{*}\boldsymbol{D}\boldsymbol{H} + \sum_{i=1}^{N_t} \{K_{1i} [\, Re(\boldsymbol{X}_i^{*}\boldsymbol{H}) - u_i \,]^2 \} + K_2 \sum_{i=1}^{N_t} [\, Im(\boldsymbol{X}_i^{*}\boldsymbol{H}) \,]^2 . \qquad (3.13)$$

3.2.4. Selection of u_i's

One issue that must be addressed here is the selection of the user-specified correlation peaks at the origin of the output correlation plane for each training image. These are the u_i values in Eq. (3.13). Since the filter **H** is constrained to a particular SLM characteristic, there is a maximum limit to these output correlation values. For a single training image this correlation value is given by Eq. (3.4). It can be written in the one-dimensional continuous notation as

$$I = \int X^*(u)\, H(u)\, du = A\, e^{j\alpha}, \tag{3.14}$$

where

$$X(u) = \mathcal{F}\{x(t)\} = |X(u)|\, e^{j\theta_x(u)},$$

and

$$H(u) = \mathcal{F}\{h(t)\} = |H(u)|\, e^{j\theta_h(u)} \tag{3.15}$$

represent the image and filter respectively. The objective here is to maximize I in Eq. (3.14) when the values of H(u) are constrained to lie on the SLM operating curves (for example, as in Figure 4.4). The method presented here follows the technique suggested by Farn and Goodman [10].

Consider the equation

$$A(\alpha) = Re\left[\left(\int X^*(u)\, H(u)\, du\right) e^{-j\alpha}\right]$$
$$= \int Re\left[X^*(u)\, H(u)\, e^{-j\alpha}\right] du \qquad (3.16)$$
$$= \int |X(u)|\, |H(u)| \cos\left[\theta_H(u) - \theta_X(u) - \alpha\right] du.$$

Here Eq. (3.16) is to be maximized over the region of realizability of H, that is, the SLM constraint. Thus

$$\max_{H(u)} A(\alpha) = \max_{H(u)} \int |X(u)|\, |H(u)| \cos\left[\theta_H(u) - \theta_X(u) - \alpha\right] du. \qquad (3.17)$$

Since H(u) can be chosen independently for each frequency u, Eq. (3.17) can be written as

$$\max_{H(u)} A(\alpha) = \int |X(u)| \max_{H(u)} \left\{|H(u) \cos\left[\theta_H(u) - \theta_X(u) - \alpha\right]\right\} du. \qquad (3.18)$$

Calculating Eq. (3.18) to determine the maximum correlation value directly will be very computation intensive. The procedure can be simplified as follows. Consider the intermediate function defined as

$$G(\Phi) = \max_{H(u)} \left\{ Re\left[H(u)\, e^{-j\Phi}\right]\right\}$$
$$= \max_{H(u)} \left\{|H(u)| \cos\left[\theta_H(u) - \phi\right]\right\}. \qquad (3.19)$$

The function G can be computed in advance as per Eq. (3.19) since the region of realizability of the filter H (namely, the

SLM constraint) is known. Further G is periodic. Substituting Eq. (3.19) in Eq. (3.18) gives

$$\max_{H(u)} A(\alpha) = \int |X(u)|\, G[\alpha + \theta_X(u)]\, du. \tag{3.20}$$

It is much easier to calculate Eq. (3.20) since the function G is already known.

The above equations can be written in discrete form. In the following equations the overbar notation represents the discrete form of the respective signals. The constraint function is discretized by sampling its phase into N values with increments of δ, where N is an integer.

Thus

$$\delta = 2\pi/N. \tag{3.21}$$

The phase-magnitude tabular data used in the optimization procedure of the SLM-MACE filter (as explained in Section 3.2) can be used here. The discrete version of Eq. (3.19) can be written as

$$\overline{G}(n) = \max_{\overline{H}(k)} \{ |\overline{H}(k)| \cos[(n_H(k) - n)\delta] \}, \quad n = 0, 1, \ldots, N-1, \tag{3.22}$$

where n_H can only take on integer values. In Eq. (3.22) the $\overline{H}(k)$ values are chosen from the phase-magnitude data mentioned above. Thus the values for $\overline{G}(n)$ can be computed as per Eq. (3.22). The discrete version of Eq. (3.20) can be written as

below. The phase values of the sampled image $\overline{X}(k)$ are quantized to the value $n_x\delta$, where n_x represents integer values. Thus

$$\overline{A}(n) = \sum_{k=0}^{d-1} |\overline{X}(k)|\,\overline{G}[n-n_x(k)], \quad n = 0, 1, \ldots, N-1, \quad (3.23)$$

where d is the total number of pixels in the digitized image $\overline{X}(k)$. Since the $\overline{G}(n)$ values are already known and it is periodic, computing the values for the function $\overline{A}(n)$ is much easier as opposed to solving Eq. 3.18. The maximum value of the function $\overline{A}(n)$ gives the maximum possible correlation value at the origin for the image x(t) if the filter is constrained to lie on a given SLM operating curve.

3.3. Summary

In this chapter the development of the SLM-Constrained MACE Filter was discussed along with the simulated annealing technique of optimization which was used in the filter design. The algorithm used for the synthesis of the filter was also presented. In the following chapter the distortion-tolerance performance of the SLM-Constrained MACE filter is evaluated with the help of computer simulation results. The performance of the filter is also evaluated in the presence of background noise in the input image.

CHAPTER 4

Performance Evaluation of the SLM-MACE Filter

In Chapter 3 the development of the SLM-Constrained MACE Filter was discussed. An algorithm for its construction was also presented. In this chapter the distortion tolerance properties of the SLM-Constrained MACE filter and its performance in the presence of input noise are studied using computer simulations.

4.1. Simulation of the SLM-Constrained MACE Filter

The SLM-Constrained MACE filter was simulated on a computer using the algorithm described in Section 3.2.3 of Chapter 3. Henceforth this filter is referred to as the SLM-MACE filter. The filter was synthesized to discriminate between two images consisting of the landscape of Mars. These images are referred to as mars1 and mars2 and are shown in Figures 4.1 and 4.2. The two images are classified as belonging to Class I and Class II, respectively. These two images of pixel size 32x32 are padded with zeroes to generate 64x64 pixel images. The SLM-MACE filter is simulated to discriminate not only between the two reference images mars1 and mars2 but also between their distorted versions. The distortions considered here are in-plane rotations.

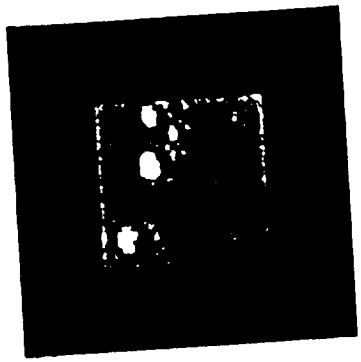

Figure 4.1. The Image mars1

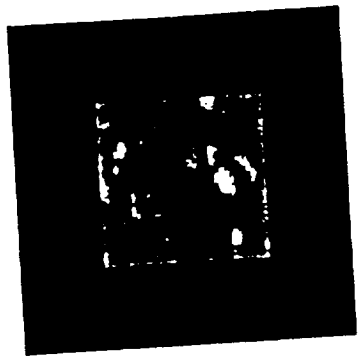

Figure 4.2. The Image mars2

The SLM-MACE filters were constructed using three different constraints. Figures 4.3 and 4.4 show the first two constraints referred to as CON1 and CON2 respectively. The figures illustrate the magnitude-phase relationship that should be maintained for the filter in order for it to be implemented on an SLM with these constraints. The third constraint considered was the Phase-Only Filter (POF) type. For this constraint the magnitude is unity for all phase values. The SLM-MACE filters were synthesized for the three SLM constraints to address the two-class pattern recognition problem mentioned earlier. The training set images for the filters were evenly distributed within the angle of rotation of the two images.

The database used to test the recognition (classification) and distortion tolerance of the filters consisted of nine images (training as well as non-training) of each class. The nine images were obtained by rotating the images mars1 and mars2 through angles zero to eight degrees. Besides their ability to distinguish between the two classes, the filters are evaluated on the basis of their correlation plane energy and the peak-to-sidelobe ratio (PSR) in the correlation plane. The threshold used for separating Class I from Class II was

$$T = 0.25\,u_1 + 0.75\,u_2\,, \qquad (4.1)$$

where u_1 and u_2 are the user-specified values for the filter

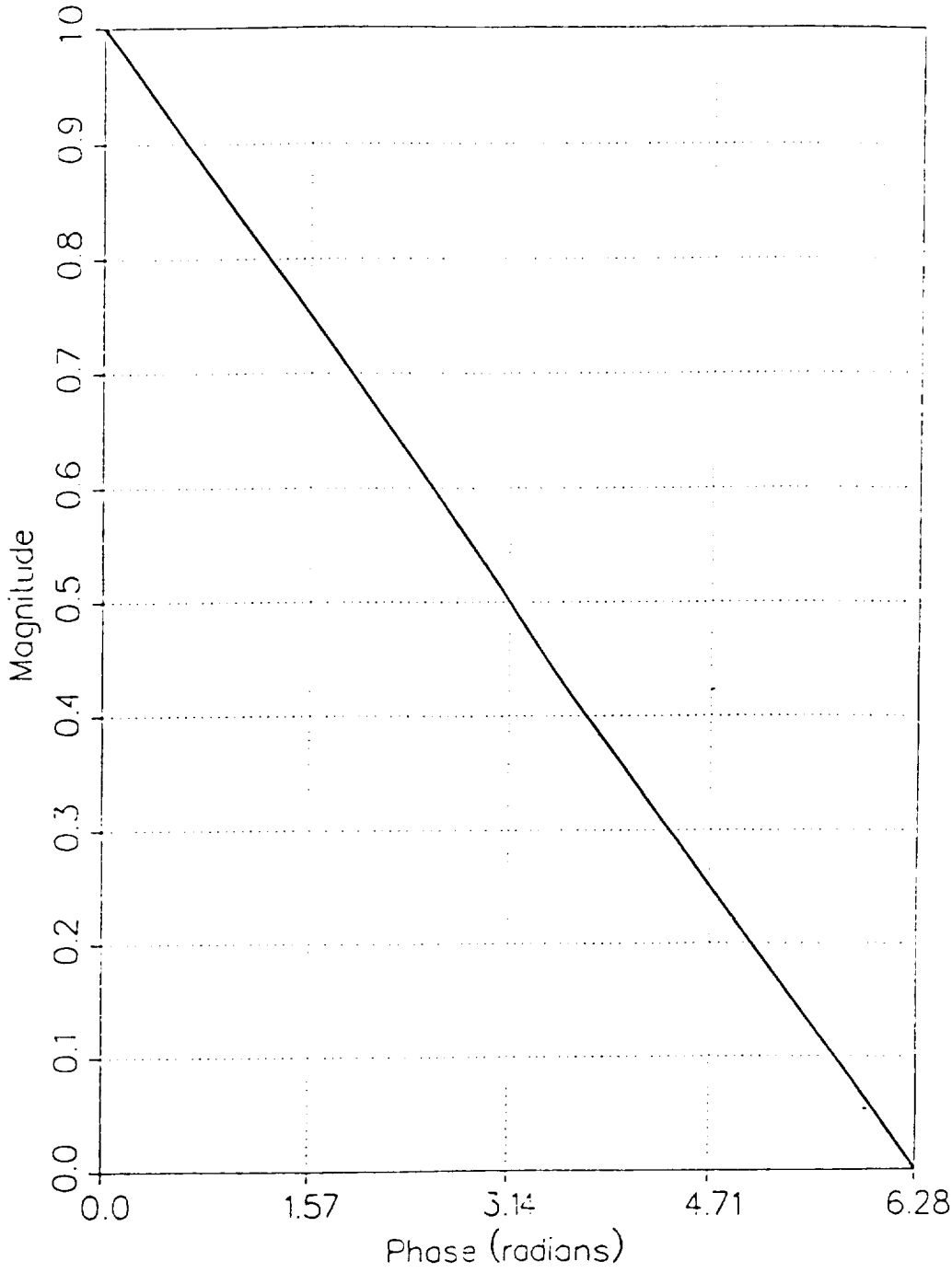

Figure 4.3. Magnitude–Phase Relationship for the Constraint CON1

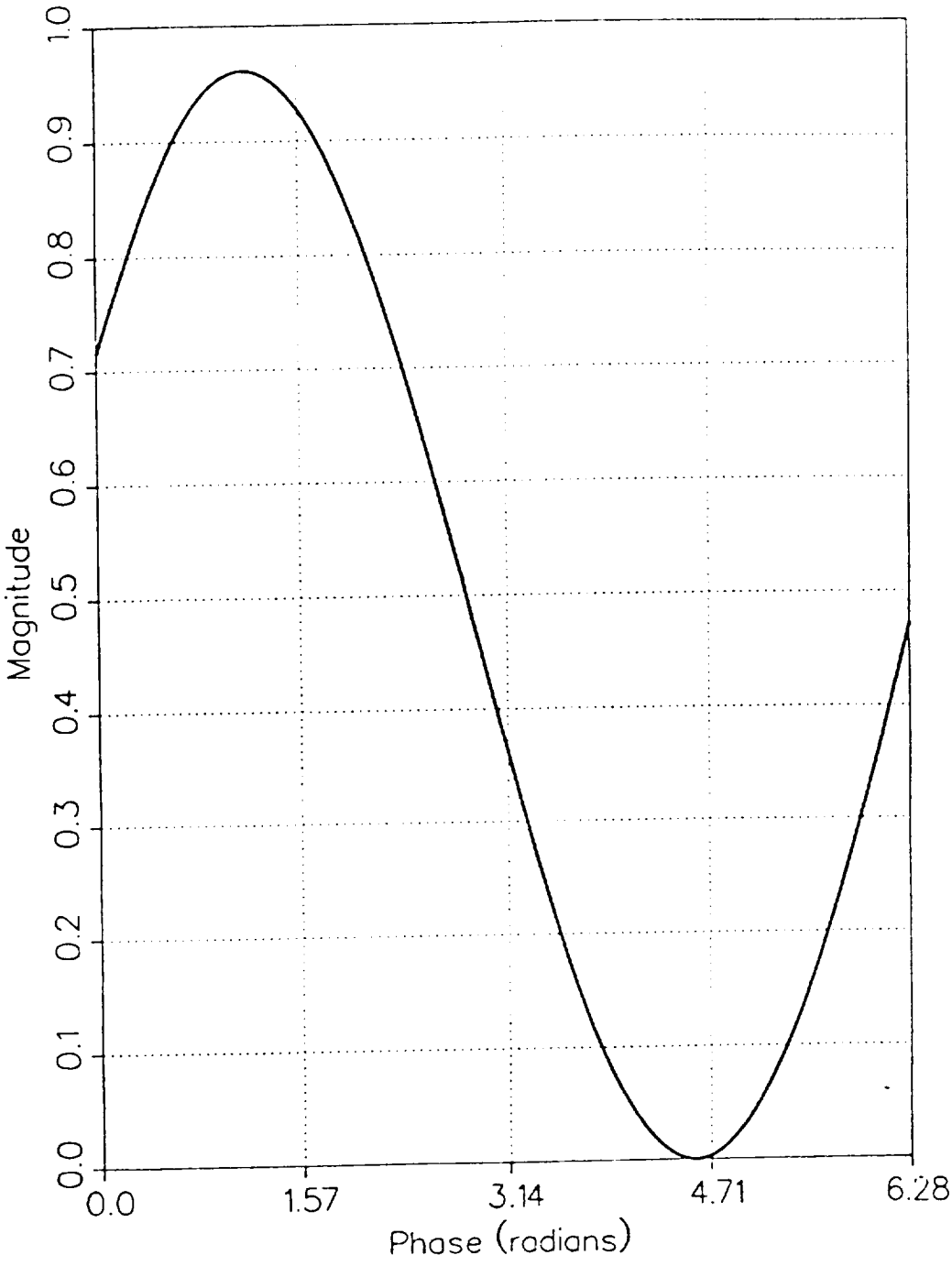

Figure 4.4. Magnitude–Phase Relationship for the Constraint CON2

response at the origin of the correlation plane for Class I and Class II images, respectively. The weight for Class I was kept low since $u_1 > u_2$ by choice. The pattern classification problem can be extended to the multiple classes case as well. In that case the number of thresholds required to discriminate between the different classes would be more than one. In the next section the distortion test results for the SLM-MACE filter are presented.

Two other filters were simulated on the computer for comparison with the SLM-MACE filter. One is the analytical MACE as discussed in Section 2.1 of Chapter 2. The MACE filter so obtained has arbitrary complex values and thus cannot be implemented on practical SLMs. To implement the analytic MACE filter on a SLM with known operating characteristics, it was modified as follows. At each frequency the phase value of the MACE filter was retained while its magnitude value was changed as per the phase-magnitude relationship of the constraints (CON1, CON2, or the POF type). This modified MACE filter is referred to as the Mod-MACE filter. The distortion test results of the SLM-MACE filter are compared to that of the MACE and Mod-MACE filters.

4.2. Distortion Test Results

The distortion-test results for the SLM-MACE filter along with that of the MACE and the Mod-MACE filters are presented in this section. All the filters were correlated with both training and non-training images of Class I and Class II. The peak value from the resulting correlation plane was compared

to the threshold value T (Eq. 4.1). If the resulting correlation peak value was greater than T, then the input image was said to belong to Class I, otherwise it was classified as a Class II image.

4.2.1. Initial Test Results

In the initial tests the training set for the filter synthesis consisted of two images from Class I and Class II, respectively. The training images for Class I consisted of two rotated versions of mars1, where the angles of rotation were zero and eight degrees. Similarly, the training images for Class II were obtained from the image mars2. Using these four training images three SLM-MACE filters were synthesized for the constraints CON1, CON2, and POF.

Now the values at the origin of the output correlation plane should be specified. As discussed in Section 3.2.4 of Chapter 3 it is important to know the maximum permissible output correlation peak values for each of the training images for the three constraints. Thus the specified peaks should be smaller than these maximum values. Using Eq. (3.23) the maximum correlation values at the origin were computed and are summarized in Table 4.1 below. The maximum correlation values for the two extreme images from each class are shown in the table. The corresponding values for the other images in the database lie within the two extreme values shown for each class. Thus, the specified correlation peak values must be less than the least value shown in Table 4.1 for each class.

Table 4.1. Maximum Correlation Values at the Origin

Image	Constraint CON1	Constraint CON2	Constraint POF
mars1.0	1.978E+06	1.803E+06	2.921E+06
mars1.8	2.456E+06	2.226E+06	3.583E+06
mars2.0	1.902E+06	1.731E+06	2.798E+06
mars2.8	2.372E+06	2.146E+06	3.449E+06

Correlation peak amplitudes of 0.80E+06 and 0.40E+06 were specified for Class I and Class II images, respectively, for the cases of the constraints CON1 and CON2. Correlation peak amplitudes of 2.1E+06 and 1.05E+06 were specified for Class I and Class II images for the case of the POF constraint. Ideally one would specify a value of 0.0 for Class II images so that there is sufficient difference between the values of the two classes. This would facilitate maximum discrimination between the two Classes. However, if the correlation value at the origin is specified to be 0.0 then the sidelobes in the correlation plane will be of values greater than 0.0. This would result in an error in detection if the detector at the output of the optical correlator is such that it looks for the peak to determine the presence of a required image. To avoid this the correlation peaks for the Class II images were specified to be half those of Class I images. The MACE filter and the Mod-MACE filter (for the constraints CON1, CON2 and POF) were also synthesized with the above mentioned training

set images. The user-specified correlation peak amplitudes for these filters were the same as those specified for the corresponding SLM-MACE filters.

The average correlation plane energy for the training images is given by: $E_{av} = \boldsymbol{H}^{*}\boldsymbol{D}\boldsymbol{H}$. The E_{av} values obtained for the SLM-MACE filters (CON1 and CON2) were 2.89E+09 and 1.23E+09 respectively. These values are relatively large compared to the E_{av} of 3.32E+08 obtained for the corresponding MACE filter. The energies of 5.37E+10 and 1.482E+10 were obtained with the Mod-MACE filters for the constraints CON1 and CON2 respectively. These values are much higher than those obtained for the corresponding SLM-MACE filters. The energy obtained with the POF SLM-MACE filter was 4.18E+10 which is much higher than the corresponding energy of 2.29E+09 obtained with the MACE filter. With the Mod-MACE (POF) filter the E_{av} obtained was 5.37E+10 which is much higher than that of the corresponding SLM-MACE filter.

The average correlation plane energy of the training images for a filter indicates its performance. A low value for the E_{av} signifies a higher peak-to-sidelobe ratio (PSR), sharper correlation peaks, and consequently more sensitivity to distortions in the test image. This is because sharp peaks are like delta functions with sharp fall-offs. Hence, if the training images for the filter are far apart then correlation peaks of the filter with the non-training images are also expected to fall sharply. As observed in the previous

paragraph, for a given set of training images the MACE filter produced the smallest E_{av} and the Mod-MACE the largest. The SLM-MACE filter achieved an E_{av} of intermediate value. Then, based on the above comments the MACE filter should be more sensitive to the non-training images of a given class than the SLM-MACE filter.

Figure 4.5 shows a typical three-dimensional plot of the output correlation plane for the SLM-MACE filter. Figures 4.6 and 4.7 show the correlation plane values along the X and Y axes, respectively. It can be seen from these two figures that the peak occurs at the origin of the correlation plane, namely, position (33,33). Similar three-dimensional plots for the MACE and Mod-MACE filters are shown in Figures 4.8 and 4.9, respectively. As observed in the plots, the correlation peak is very sharp and the sidelobe suppression is excellent in case of the MACE filter. In case of the SLM-MACE filter the sidelobes are not as low as the MACE filter, but the peak is sharp enough relative to the sidelobes, facilitating easy detection. The correlation surface of the Mod-MACE filter is a little poorer compared to the other two with respect to the sidelobe suppression. All the three filters seem to be fairly good in their correlation surface characteristics. It is to be seen how they compare in distortion invariance and classification.

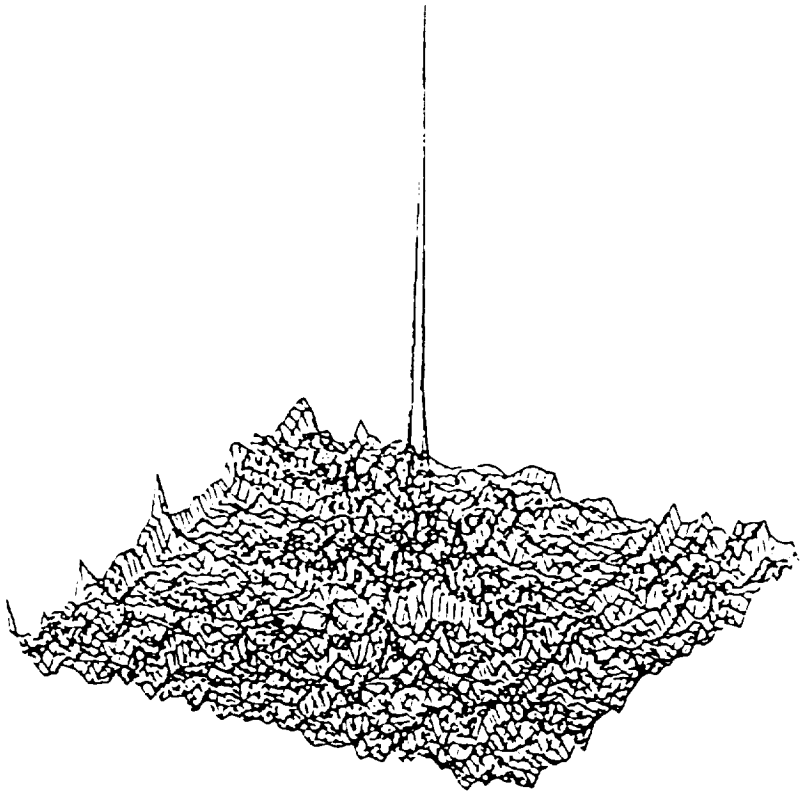

Figure 4.5. The Correlation Plane Output for SLM-MACE Filter
Correlation Peak = 0.8E+06
Correlation Plane Energy = 1.173E+09

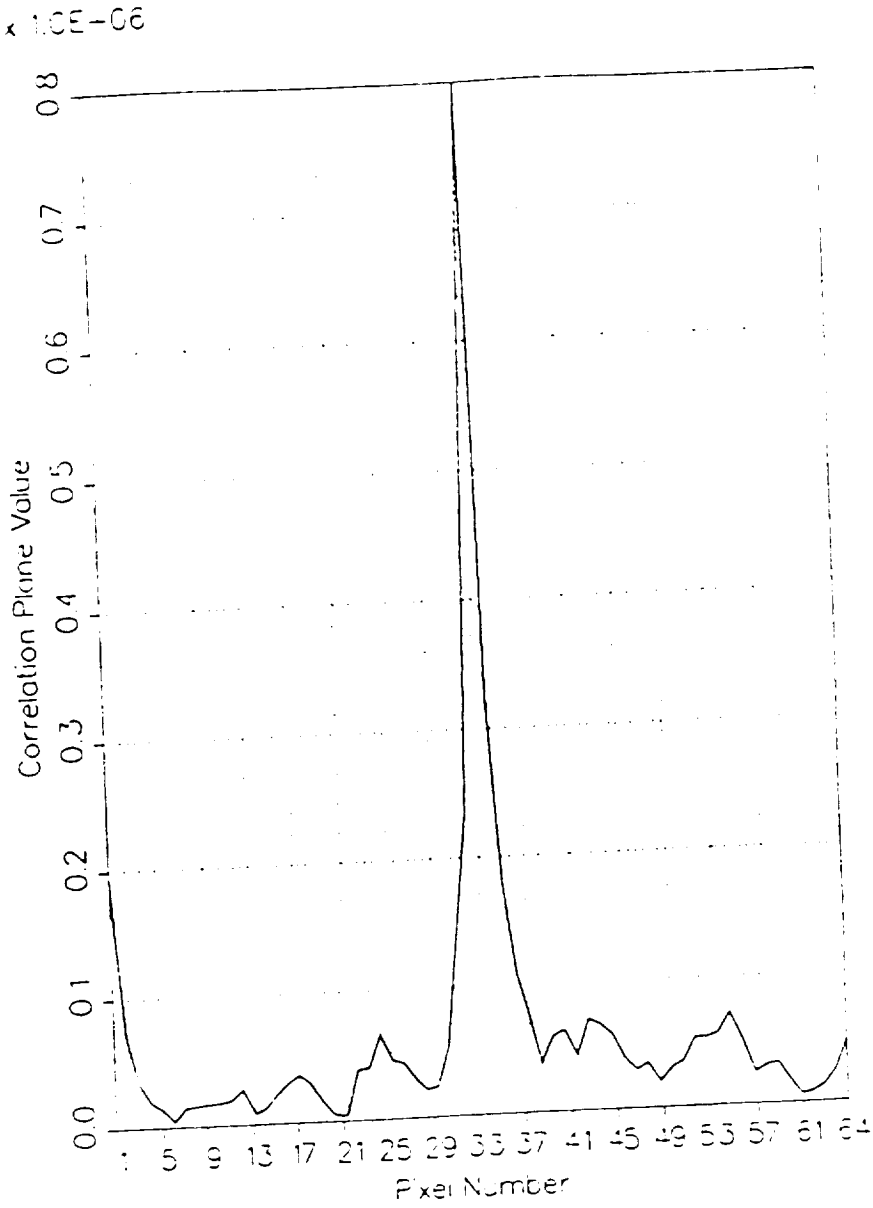

Figure 4.6. Correlation Plane Values Along X-Axis (Through Origin) for SLM-MACE Filter

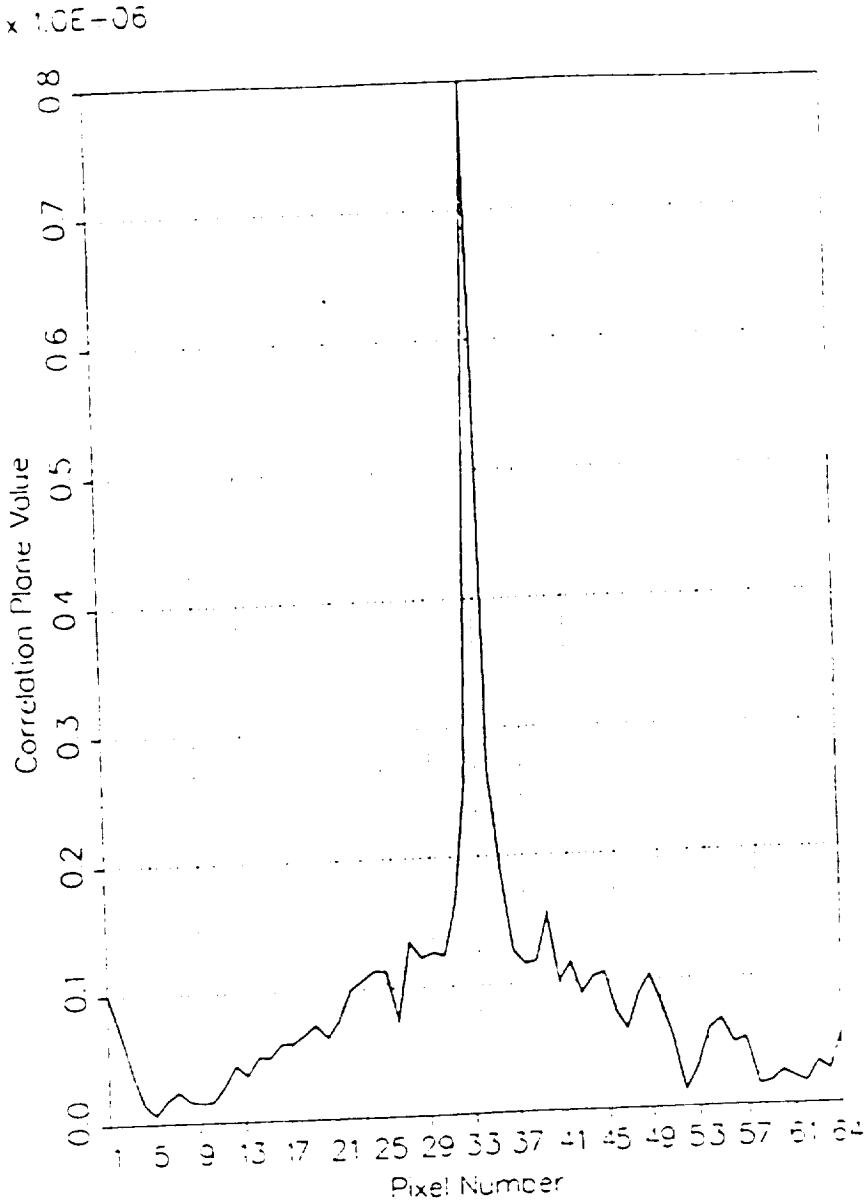

Figure 4.7. Correlation Plane Values Along Y-Axis (Through Origin) for SLM-MACE Filter

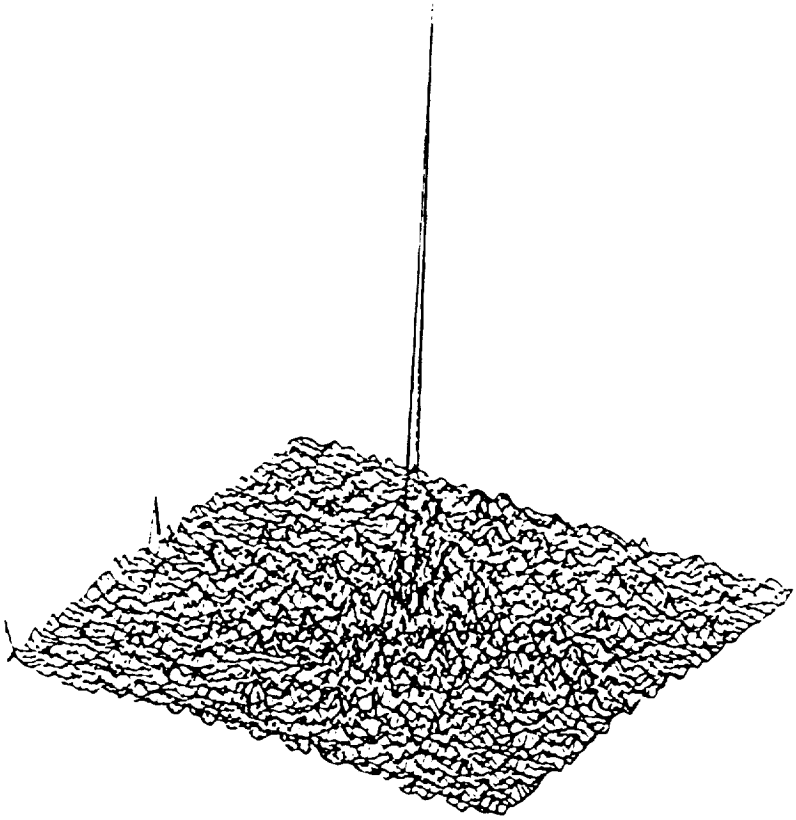

Figure 4.8. The Correlation Plane Output for a MACE Filter
Correlation Peak = 0.8E+06
Correlation Plane Energy = 3.727E+08

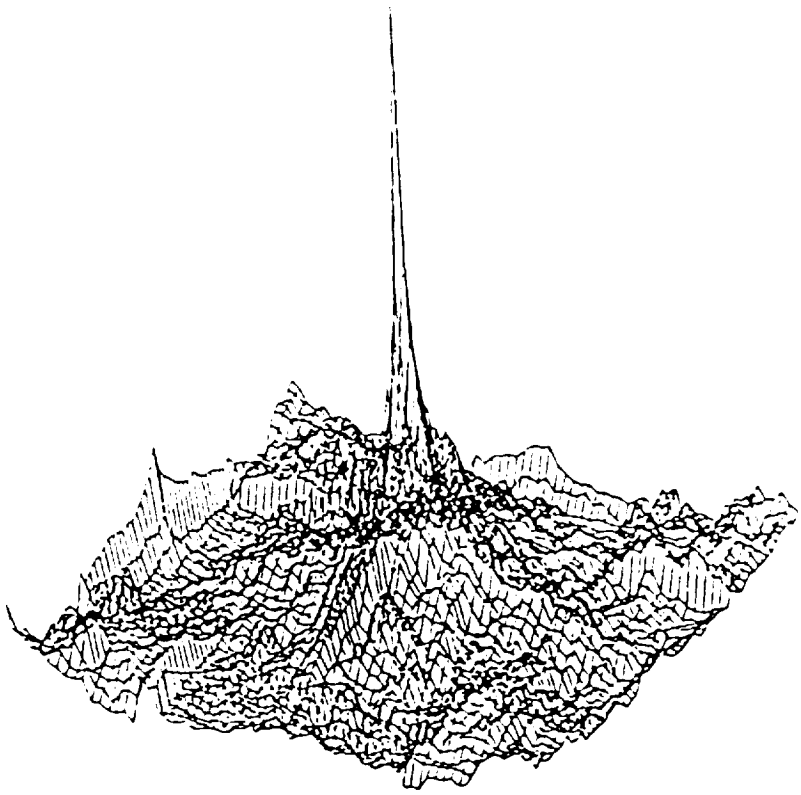

Figure 4.9. The Correlation Output Plane for a Mod-MACE Filter
Correlation Peak = 1.156E+06
Correlation Plane Energy = 1.062E+10

The three SLM-MACE filters (synthesized for the three constraints) and the corresponding MACE and Mod-MACE filters were correlated with the training and non-training images of Class I and Class II. Tables 4.2, 4.3, and 4.4 list the correlation plane statistics for the Class I and Class II training images for the SLM-MACE filters. For each tested image the data include the specified value at the origin of the correlation plane, the largest observed peak in the correlation plane, the peak-to-sidelobe ratio (PSR) and the correlation plane energy. In the case of all the tested images the largest peak in the correlation plane occurred at the origin. As seen from the tables, for the training images used in the three SLM-MACE filters, the imposed constraints at the origin of the correlation plane were satisfied. Also the PSR values are reasonably high indicating small sidelobes.

Table 4.2. Correlation Plane Statistics for SLM-MACE Filter (using Constraint CON1)

Test Image	Specified Intensity at the origin	Largest Peak Observed	PSR	Corr. Plane Energy
mars1.0	0.800E+06	0.80E+06	5.210	2.375E+09
mars1.8	0.800E+06	0.80E+06	3.743	3.785E+09
mars2.0	0.400E+06	0.40E+06	1.903	2.079E+09
mars2.8	0.400E+06	0.40E+06	2.071	3.346E+09

Table 4.3. Correlation Plane Statistics for SLM-MACE Filter (using Constraint CON2)

Test Image	Specified Intensity at the origin	Largest Peak Observed	PSR	Corr. Plane Energy
mars1.0	0.800E+06	0.800E+06	4.115	1.173E+09
mars1.8	0.800E+06	0.800E+06	3.534	1.573E+09
mars2.0	0.400E+06	0.400E+06	1.824	9.434E+08
mars2.8	0.400E+06	0.400E+06	2.399	1.242E+09

Table 4.4. Correlation Plane Statistics for SLM-MACE filter (Using POF Constraint)

Test Image	Specified Intensity at the origin	Largest Peak Observed	PSR	Corr. Plane Energy
mars1.0	2.10E+06	2.10E+06	3.466	3.432E+10
mars1.8	2.10E+10	2.10E+10	2.567	5.434E+10
mars2.0	1.05E+06	1.05E+06	1.620	3.030E+10
mars2.8	1.05E+06	1.05E+06	1.323	4.820E+10

The same correlation plane statistics for the MACE filter and the Mod-MACE filters (for the three constraints) are shown in Tables 4.5, 4.6, 4.7, and 4.8, respectively. Again for all these filters, the largest peak observed in the correlation plane occurred at the origin. The correlation plane energies for the MACE filter are much lower than the SLM-MACE filters and consequently the PSR values are higher. In the case of the Mod-MACE filter the PSR values are lower than the SLM-MACE and MACE filters. From Table 4.5 it can be seen that the correlation peaks obtained from the images used in the MACE filter construction satisfied the imposed constraints at the origin. However, the same is not true for the Mod-MACE filters as observed in Tables 4.6, 4.7, and 4.8. Here the correlation peaks are much higher than the values specified in the imposed constraints. This will affect its recognition capability as will be seen later on in this chapter.

Table 4.5. Correlation Plane Statistics for the MACE filter

Test Image	Specified Intensity at the origin	Largest Peak Observed	PSR	Corr. Plane Energy
mars1.0	0.80E+06	0.80E+06	7.232	3.727E+08
mars1.8	0.80E+08	0.80E+08	7.393	3.856E+08
mars2.0	0.40E+06	0.40E+06	3.426	2.746E+08
mars2.8	0.40E+06	0.40E+06	3.881	2.961E+08

Table 4.6. Correlation Plane Statistics for Mod-MACE filter (Using Constraint CON1)

Test Image	Specified Intensity at the origin	Largest Peak Observed	PSR	Corr. Plane Energy
mars1.0	0.80E+06	2.330E+06	2.859	3.821E+06
mars1.8	0.80E+06	2.924E+06	2.963	6.054E+10
mars2.0	0.40E+06	1.634E+06	2.170	3.374E+10
mars2.8	0.40E+06	2.121E+06	2.294	5.372E+10

Table 4.7. Correlation Plane Statistics for Mod-MACE filter (Using Constraint CON2)

Test Image	Specified Intensity at the origin	Largest Peak Observed	PSR	Corr. Plane Energy
mars1.0	0.80E+06	1.156E+06	2.881	1.062E+10
mars1.8	0.80E+06	1.456E+06	2.962	1.672E+10
mars2.0	0.40E+06	0.809E+06	2.192	0.936E+10
mars2.8	0.40E+06	1.054E+06	2.261	1.482E+10

Table 4.8. Correlation Plane Statistics for Mod-MACE filter (Using Constraint POF)

Test Image	Specified Intensity at the origin	Largest Peak Observed	PSR	Corr. Plane Energy
mars1.0	2.10E+06	2.330E+06	2.865	3.432E+10
mars1.8	2.10E+06	2.924E+06	2.970	5.434E+10
mars2.0	1.05E+06	1.634E+06	2.172	3.030E+10
mars2.8	1.05E+06	1.121E+06	2.298	4.820E+10

The correlation test results for the SLM-MACE filter for the training and non-training images along with those for the MACE and Mod-MACE filters are summarized in Table 4.9 as Test Numbers 1 through 7. The training images used for each class for all the filters were the base images mars1 and mars2 rotated through zero and eight degrees.

In order to classify the test images between the two classes the threshold T was chosen as 0.5E+06 (from Eq. 4.1 with u_1 = 0.8E+06 and u_2 = 0.4E+06) for the SLM-MACE and Mod-MACE filters (with constraints CON1 and CON2). The same threshold was used for the MACE filter. In the case of the POF Constraint, T = 1.3125E+06 (since u_1 = 2.1E+06 and u_2 = 1.05E+06) for the SLM-MACE and Mod-MACE filters. As seen in Table 4.9 the SLM-MACE filters recognized the test images correctly on every occasion. However, in the case of the MACE filter some non-training images from Class I were misclassified giving a recognition of 56 percent. This is because the MACE filter gives sharper correlation peaks than the SLM-MACE filter making it sensitive to non-training images of a given class. Thus the performance of the SLM-MACE filter was better than the MACE and Mod-MACE filters.

A graphical representation of these results is given in Figures 4.10-4.14. In each of these figures the global correlation peaks are plotted against their respective image numbers. The Class I images are represented with a o and the Class II images are represented with a □.

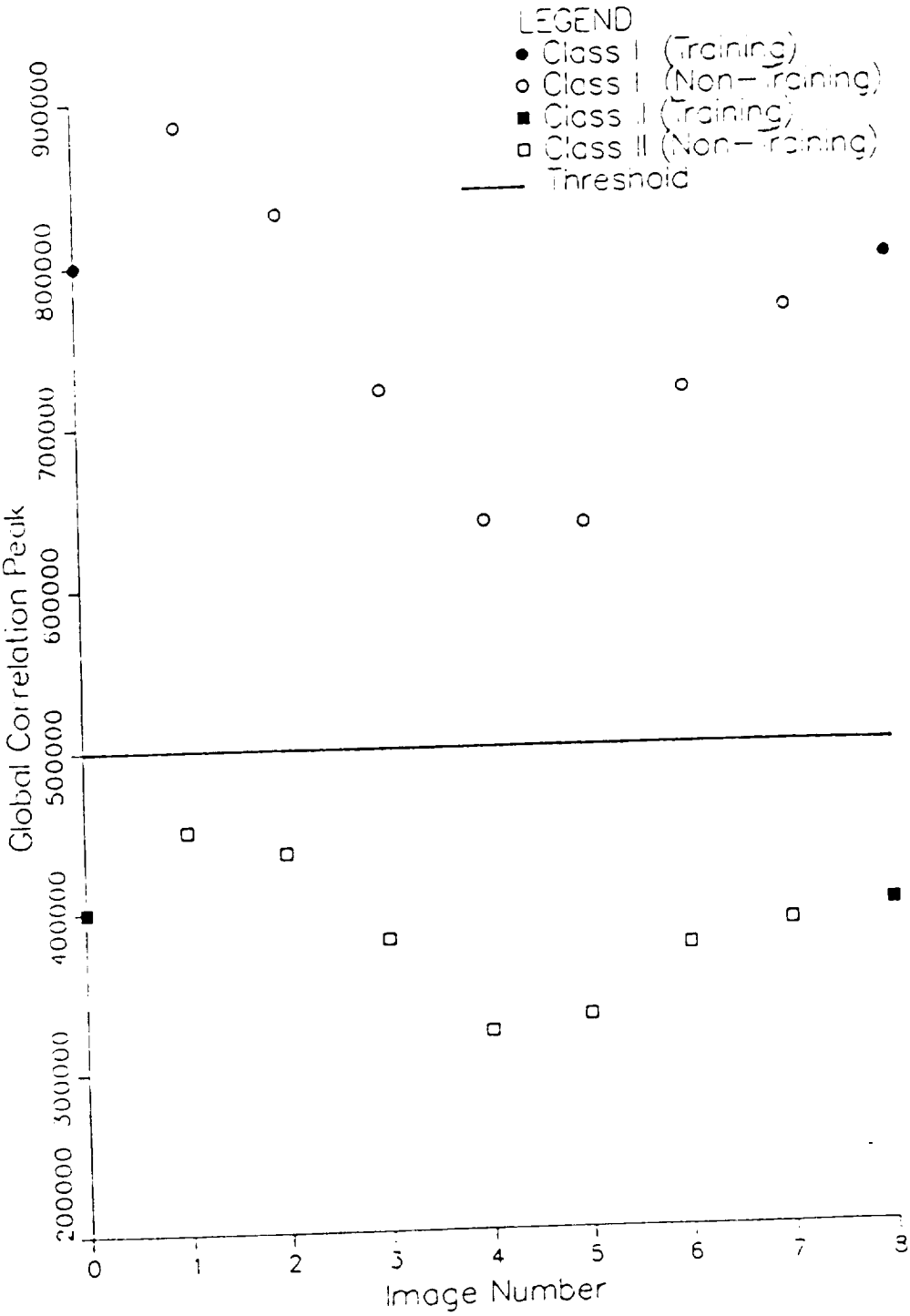

Figure 4.10. Distortion Test Results for the SLM-MACE Filter (using Training Images 3 and 8) for the Constraint CON1

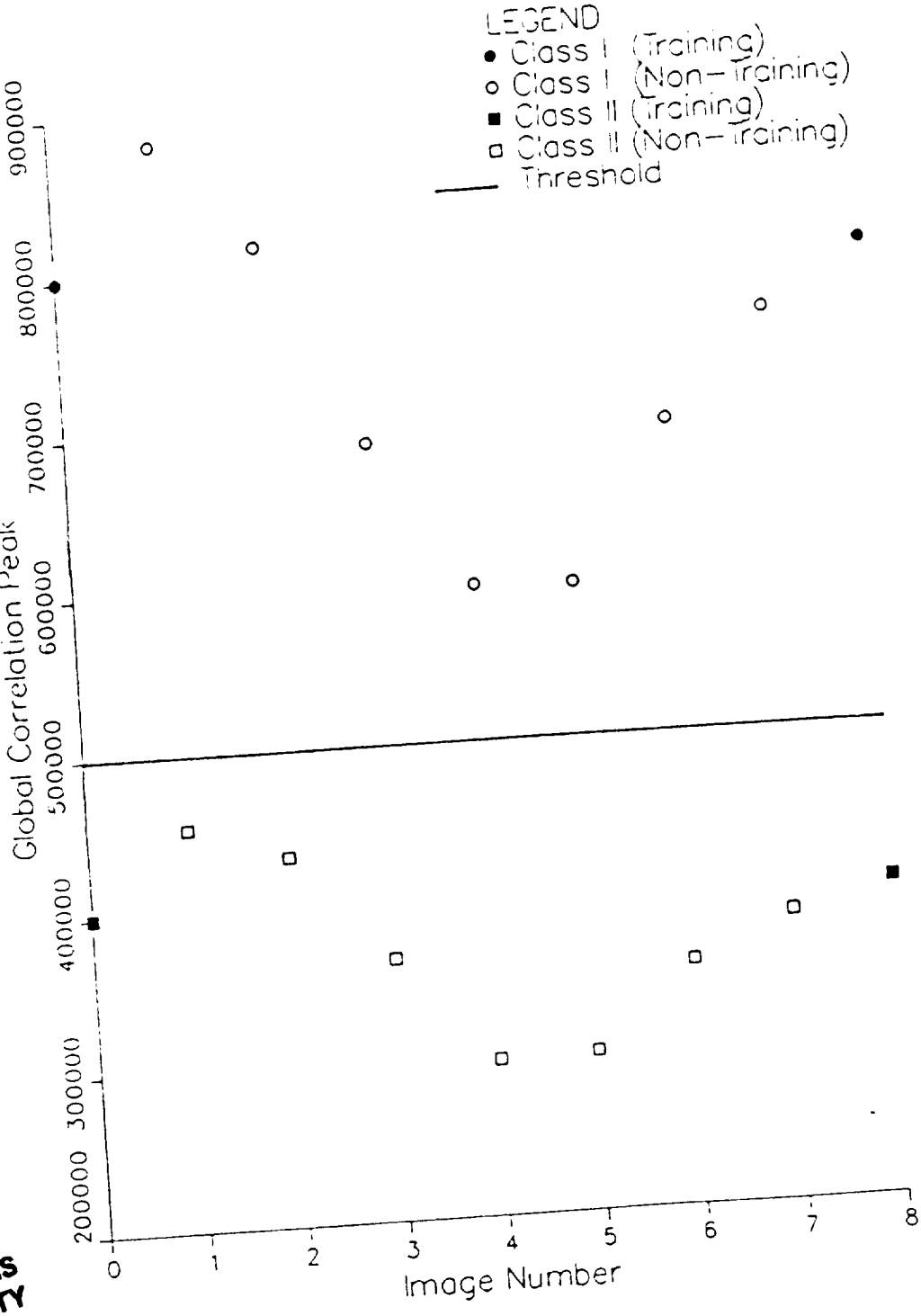

Figure 4.11. Distortion Test Results for the SLM-MACE Filter (using Training images 0 and 8) for the Constraint CONS

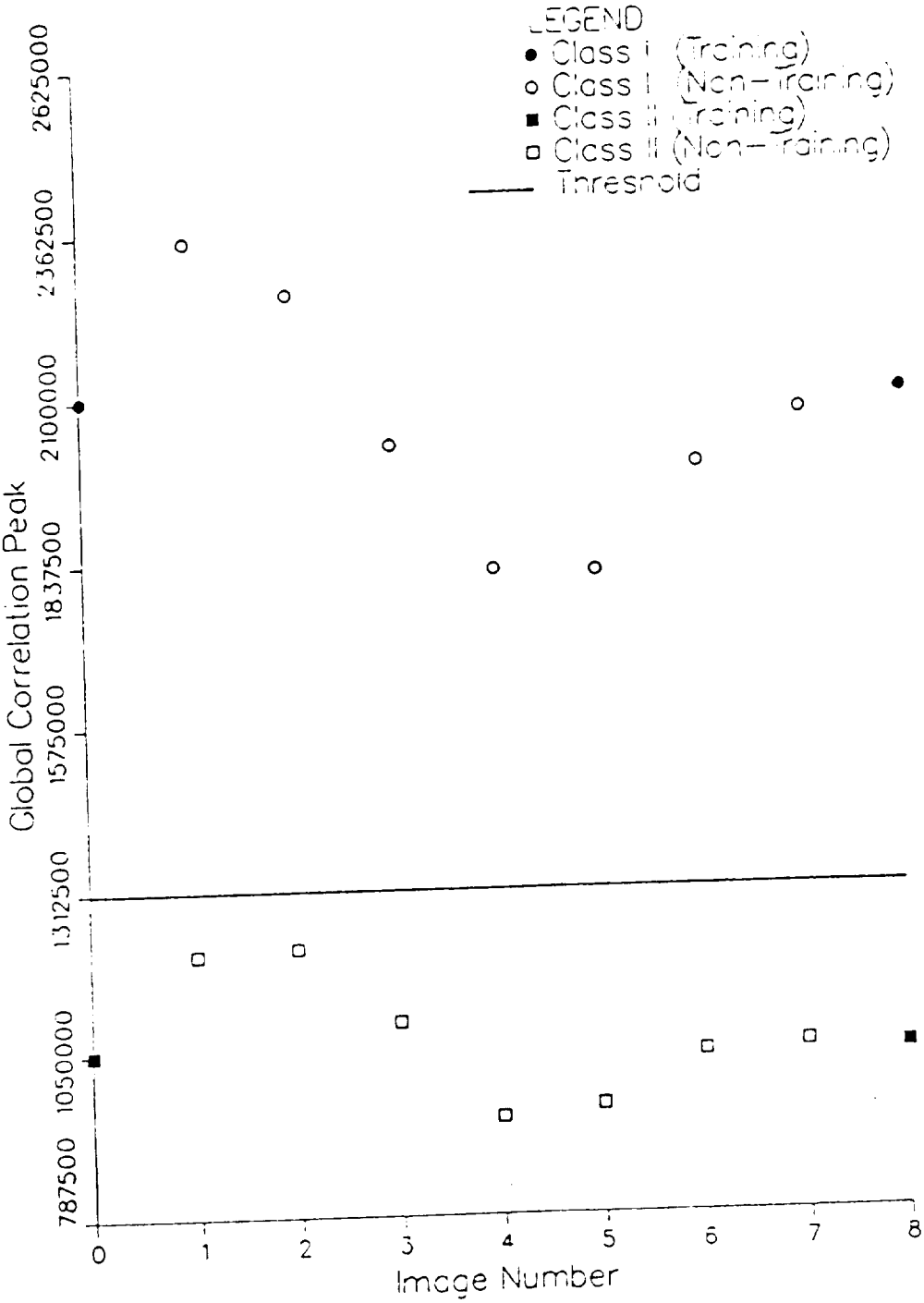

Figure 4.12. Distortion Test Results for the SLM-MACE Filter (using Training Images 0 and 8) for the POF Constraint

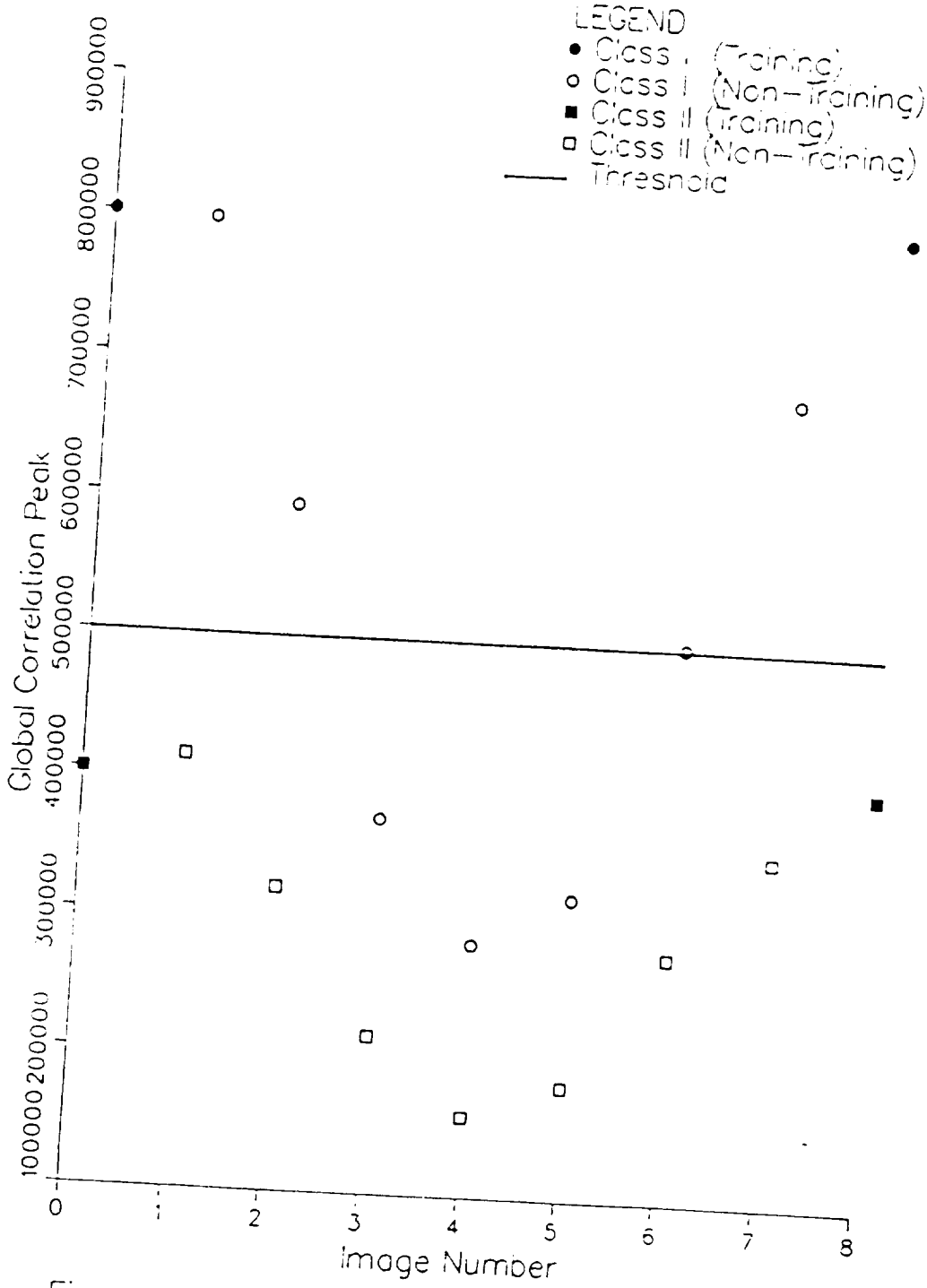

Figure 4.13. Distortion Test Results for the MACE Filter (using Training Images 0 and 5)

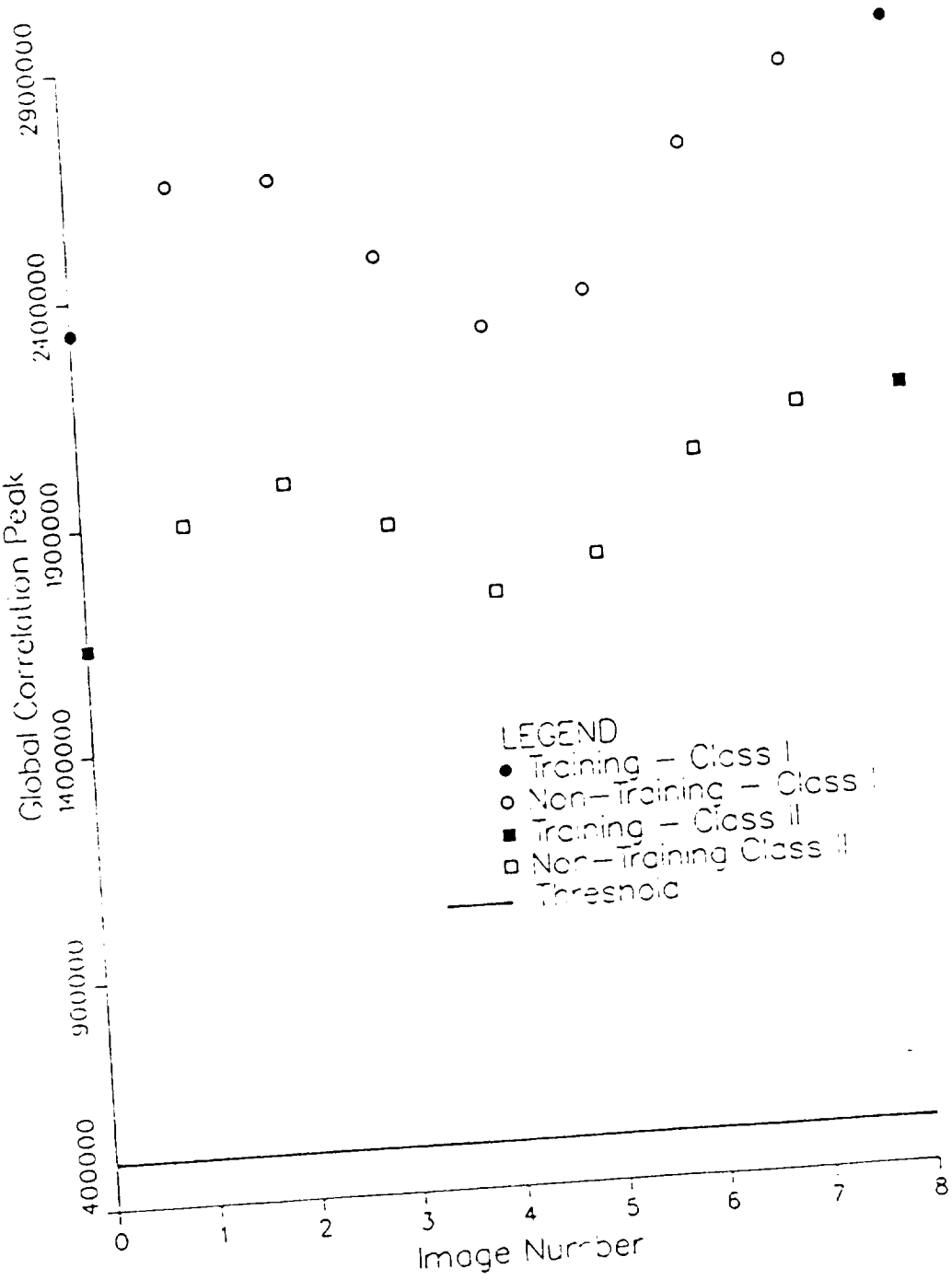

Figure 4.14. Distortion Test Results for the Mod–MACE filter (using Training Images 0 and 3) for Constraint CON1

The solid symbols denote the images used in the training set for the two classes. The solid line denotes the threshold level used for classifying the images. If the global correlation peak for an image falls above this line then it is classified as a Class I image, otherwise it is a Class II image. As opposed to the SLM-MACE and MACE filters the imposed constraints at the origin of the correlation plane are not met in the case of the training images of the Mod-MACE filter. As seen in Figure 4.14 this resulted in all the correlation peaks falling above the threshold level for the Class II images for this filter. Hence it achieved a recognition rate of 0 for Class II.

4.2.2. Further Tests with Increased Number of Training Images

As seen from the earlier test results, the distortion tolerance of the MACE and Mod-MACE filters is not very satisfactory. A way of improving the distortion tolerance capability of these filters to the non-training images is to increase the number of training images within the angles of rotation. Tests 1-7 were repeated with three training images for each class. These three images were the base image of each class rotated through angles zero, four, and eight degrees, respectively. The results of these tests are summarized in Table 4.10 as Tests 8-14. As seen in Table 4.10, increasing the number of training images resulted in 100 percent recognition for both the Classes in the case of the MACE filter (Test 11) which is an improvement over the

corresponding MACE filter from Table 4.9 (Test 4). But this is not true of the Mod-MACE filters (Tests 12-14). These filters continued to have 0 percent recognition rate for Class II images. Although the global correlation peaks do occur at the origin as desired (for the Mod-MACE filters), the imposed constraints at the origin of the correlation plane are violated leading to poor recognition.

However, there was an increase in the average correlation plane energies for all the filters in comparison to the corresponding energies from Table 4.9. This increase in energies had an uneven effect on the average PSR values for the two classes. This observation is made by comparing the average PSR values for the corresponding filters in Tables 4.9 and 4.10, respectively. To further study the effects of increasing the training set images the same seven tests were repeated with training images spaced two degrees apart (between the angles zero and eight degrees) for each Class. Thus the number of training set images for each class is increased to five per class. The results of these tests are summarized in Table 4.11 as Tests 15-21.

As seen in Table 4.11, increasing the number of training set images resulted in a further increase in the average correlation plane energies and a decrease in the average PSR values for all the filters. The recognition rate of the Mod-MACE filters was not affected by the increase in the training images. A brief summary of the distortion test results presented so far is given in the next section.

Table 4.9. Distortion Test Results for Filters Constructed with two Training Images from each Class

Test No.	Filter	Object Class	Percent Correct	Avg. PSR	Average Energy
1.	SLM-MACE (CON1)	mars1 mars2	100 % 100 %	4.066 1.931	3.605E+09 3.200E+09
2.	SLM-MACE (CON2)	mars1 mars2	100 % 100 %	3.726 1.952	1.694E+09 1.386E+09
3.	SLM-MACE (POF)	mars1 mars2	100 % 100 %	2.821 1.374	5.236E+10 4.653E+10
4.	MACE	mars1 mars2	56 % 100 %	5.455 2.619	4.571E+08 4.022E+08
5.	Mod-MACE (CON1)	mars1 mars2	100 % 0 %	2.824 2.098	5.829E+10 5.180E+10
6.	Mod-MACE (CON2)	mars1 mars2	100 % 0 %	2.849 2.117	1.620E+10 1.439E+10
7.	Mod-MACE (POF)	mars1 mars2	100 % 0 %	2.830 2.100	5.226E+10 4.653E+10

Table 4.10. Distortion Test Results for Filters Constructed with Three Training Images per Class

Test No.	Filter	Object Class	Percent Correct	Avg. PSR	Average Energy
8.	SLM-MACE (CON1)	mars1 mars2	100 % 100 %	3.876 2.013	4.042E+09 3.653E+09
9.	SLM-MACE (CON2)	mars1 mars2	100 % 100 %	4.119 1.922	2.082E+09 1.763E+09
10.	SLM-MACE (POF)	mars1 mars2	100 % 100 %	3.503 1.592	5.236E+09 4.653E+09
11.	MACE	mars1 mars2	100 % 100 %	7.084 3.086	5.086E+08 4.060E+08
12.	Mod-MACE (CON1)	mars1 mars2	100 % 0 %	3.105 2.187	5.829E+10 5.181E+10
13.	Mod-MACE (CON2)	mars1 mars2	100 % 0 %	3.135 2.198	1.620E+10 1.440E+10
14.	Mod-MACE (POF)	mars1 mars2	100 % 0 %	3.105 2.186	5.829E+10 5.181E+10

Table 4.11. Distortion Test Results for Filters Constructed with five Training Images per Class

Test No.	Filter	Object Class	Percent Correct	Avg. PSR	Average Energy
15.	SLM-MACE (CON1)	mars1	100 %	3.360	5.140E+09
		mars2	100 %	1.775	4.616E+09
16.	SLM-MACE (CON2)	mars1	100 %	3.122	3.946E+09
		mars2	100 %	1.624	4.616E+09
17.	SLM-MACE (POF)	mars1	100 %	2.366	5.236E+09
		mars2	100 %	1.126	4.653E+09
18.	MACE	mars1	100 %	6.818	4.855E+08
		mars2	100 %	2.960	3.806E+08
19.	Mod_MACE ($\overline{\text{CON1}}$)	mars1	100 %	3.118	5.829E+10
		mars2	0 %	2.184	5.181E+10
20.	Mod_MACE ($\overline{\text{CON2}}$)	mars1	100 %	3.149	1.620E+10
		mars2	0 %	2.198	1.439E+10
21.	Mod_MACE ($\overline{\text{POF}}$)	mars1	100 %	3.118	5.829E+10
		mars2	0 %	2.184	5.181E+10

4.2.3. Summary of Results

Based on the distortion test results presented earlier, it can be concluded that all the three composite filters, namely, SLM-MACE, MACE, and Mod-MACE, provide acceptable sidelobe suppression in the output correlation plane. The correlation peaks in the case of the MACE filter are very sharp and this makes it more sensitive to distortions in the non-training images compared to the SLM-MACE filter. Consequently, the MACE filter gives higher PSR values than the SLM-MACE filters. But the PSR values of the SLM-MACE filters are good enough to facilitate easy detection in the correlation plane. Thus within a given range of distortion the MACE filter requires more number of training images than the SLM-MACE filter to achieve good recognition.

Increasing the number of training images resulted in an increase in the average correlation plane energy for all the filters. This increase in the energy had varying effects on the PSR values for the two classes of images when tested with a given filter. However, the most important difference between the SLM-MACE and MACE filters is in their practical implementation. As discussed in Chapter three, currently available SLMs have limited phase and amplitude modulation capabilities, and phase-magnitude cross coupling. If the MACE filter were to be implemented on such SLMs then its recognition capabilities are very poor as observed for the Mod-MACE filter results seen earlier. However, the SLM characteristics can be incorporated into the design of the

SLM-MACE filter so that the filter is constrained to take on values within the operating constraints of a given SLM. Any arbitrary SLM constraint can be used in the design of the SLM-MACE filter which makes it more suitable for implementation in practical OPR systems.

4.3. Noise Test Results

In this section the effect of background noise (in the input test images) on the performance of the filters is studied. The distortion invariance performance of the SLM-MACE filters is evaluated in the presence of zero-mean white noise at the input. The performance of the filter is also studied in the presence of a constant bias term added to all the pixels in the test image. Similar test results for the MACE filter are presented for comparison.

4.3.1. Noise Tests

To simulate the conditions under which input images would be tested, zero-mean, white Gaussian noise was added to the images before correlating them with the filter. Noise with different signal-to-noise ratio values were added to all the images in the database (including Class I and Class II, a total of eighteen images). Figures 4.15 and 4.16 show the images mars1 and mars2 corrupted with noise. The SLM-MACE filter (using Constraint CON2) synthesized with five training set images of each Class (as in Test 16, Table 4.11) was used to test all the noisy images in the database. As before the user-specified correlation peak values for Class I and Class II were 0.8E+06 and 0.4E+06.

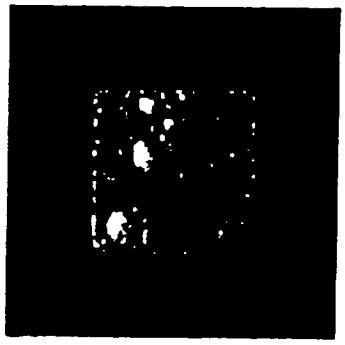

Figure 4.15. Image mars1 with noise (SNR = 10 dB)

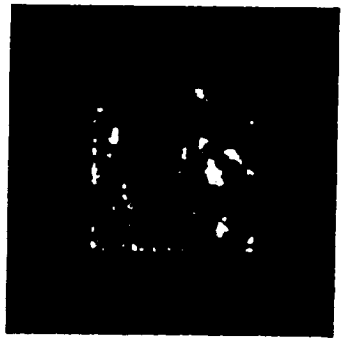

Figure 4.16. Image mars2 with noise (SNR = 10 dB)

The same threshold as in Test 16 (T = 0.5E+06) was used to classify the test images as Class I or Class II images. The results of this test for various SNR values are summarized in Table 4.12. The same tests were performed for the corresponding MACE filter (Test 18, Table 4.11). The results of these tests are summarized in Table 4.13.

As seen in Table 4.12 the SLM-MACE filter tolerates noise upto the 10 dB SNR level without any effect on its recognition capabilities. The same is true of the MACE filter as seen in Table 4.13. However, with the increase in the noise levels there is a drop in the correlation plane PSR values. For SNR values below the 10 dB level, the correlation plane value at the origin is dominated by the sidelobes resulting in poor recognition for those images.

Table 4.12. Noise Test Results for the SLM-MACE Filter (for Constraint CON1)

SNR in dB	% Recognition		PSR	
	mars1	mars2	mars1	mars2
20	100 %	100 %	3.23	1.64
15	100 %	100 %	3.03	1.67
10	100 %	100 %	3.01	1.40
5	100 %	22 %	2.62	1.38
0	100 %	67 %	1.99	1.11

Table 4.13. Noise Test Results for the MACE filter

S N R in dB	% Recognition mars	mars2	PSR mars1	mars2
20	100 %	100 %	6.20	2.54
15	100 %	100 %	4.98	2.30
10	100 %	100 %	3.63	2.01
5	100 %	11 %	1.78	1.02
0	22 %	0 %	1.10	1.14

4.3.2. Bias Correction

One other factor which should be addressed while simulating real testing conditions for the filter is the effect of input bias [37]. A constant noise term added equally to all the pixels of the input image can be considered a bias term. This will affect the correlation peak and PSR values in the correlation plane.

To correct this bias in the input image it is required that the cross-correlation of the filter and a random (but spatially constant) bias signal be zero [38]. Thus the required condition can be written as

$$h^T c = 0 , \qquad (4.2)$$

where c = [1 1 1 . . . 1] is the spatially constant image in vector

notation and h is the filter in the spatial domain. Thus the inner product of the filter and a constant image should be zero. This condition will ensure that uniform bias term added to the input image will not degrade the peak values in the correlation plane. This can be incorporated into the filter design by including a constant image in the training set and setting its corresponding correlation output to zero. To show that the above scheme helps prevent the degradation of the correlation plane PSR due to the addition of a bias term in the input image, one example situation was simulated on the computer.

An SLM-MACE filter was synthesized (using Constraint CON2) for a single reference image of mars1. The output at the origin of the correlation plane was specified as 0.8E+06. Another SLM-MACE filter was synthesized for the same image with an additional image included in the training set and its correlation output at the origin was constrained as in Eq. 4.2. Similarly, two MACE filters were synthesized (with and without the bias correction) for the same image mars1. All these four filters were then correlated with different biased versions of the training image mars1. The results of these tests are summarized in Table 4.14. The table indicates the PSR values obtained with the four filters for various bias levels added to the training image.

As seen in Table 4.14 the PSR values for the MACE filter (constructed without the bias correction) dropped steadily

Table 4.14. Correlation Test Results of the Filters with Biased Versions of the Test Image

H1 = Filter Without Bias Correction
H2 = Filter With Bias Correction

	PSR Values for			
	MACE Filter		SLM-MACE Filter (Constraint CON2)	
Bias	H1	H2	H1	H2
10.0	9040.55	4094.67	9.01436	8.86552
20.0	4521.25	4094.69	9.01434	8.86552
30.0	3014.54	4094.69	9.01432	8.86552
40.0	2261.16	4094.69	9.01430	8.86552
50.0	1809.13	4094.69	9.01428	8.86552
60.0	1507.77	4094.69	9.01426	8.86552
70.0	1292.49	4094.69	9.01424	8.86552

with increasing bias levels. This situation was rectified in the case of the MACE filter constructed with the bias correction. Here the PSR values remained steady for similar increases in the bias level. However, it was observed that in the case of the SLM-MACE filter, increasing the bias levels in the input image did not degrade its PSR values considerably. There was not much difference in the PSR values for the SLM-MACE filters constructed with and without the bias correction.

Thus it can be concluded that the SLM-MACE filter can tolerate significant bias levels in the input image while maintaining a reasonably steady PSR value.

4.4. Reducing Computation Time

As discussed in Chapter 3, the simulated annealing technique of optimization is used for the synthesis of the SLM-MACE filter. It took approximately five hours of CPU time on a SUN SPARC1+ workstation to run the computer program for the SLM-MACE filter synthesis. It would be desirable to modify this procedure in some way so that less computation time is required without affecting the distortion capabilities of the filter. In this section one such method is presented along with computer simulation results.

In real images most of the energy lies in the low frequency region of the spectrum. In Reference [39] Bahri has used square windows (centered at the origin) in the frequency domain to reject the high frequency components of the reference image. The Phase-Only filters created using this reference image was designed to yield high SNR values for detection of a single image corrupted with noise. A similar technique can be used here to ignore some of the training image pixels.

Since the filter pixels are the variables in the optimization procedure, the computation time required for the synthesis of the SLM-MACE filter (using optimization) depends on the size of the images. If only a limited number of pixels from the training images are used for the optimization then

the computation time required could be reduced. The corresponding filter pixels which are not considered for the training images can then be set to zero. This scheme was applied in the design of the SLM-MACE filter in the following manner.

A threshold is set for the magnitude of the frequency components of the training images depending upon their range of values. All training image pixels which have magnitudes below this threshold are then ignored during the filter synthesis. The threshold has to be selected such that only the low magnitude frequency components are ignored. The same algorithm as outlined in Section 3.2.3 (Chapter 3) is followed with a few modifications. In Step 5 of that algorithm, the diagonal elements of the diagonal matrix D are the average magnitude squared values of all the training images. Those diagonal elements of D which fall below a preset threshold are set to zero. The corresponding pixels in the training images are also set to zero. Now, a new diagonal matrix D is created which includes only the nonzero diagonal elements of the original D-matrix. This new diagonal matrix will have a reduced number of diagonal elements than the original matrix. Similarly, new training images are formed which include only the pixels corresponding to the nonzero diagonal elements of the original D-matrix.

The new D-matrix and training image vectors are then used in the objective function (Eq. 3.13, Chapter 3). Thus the filter $\mathbf{H}$ and the training image vectors $\boldsymbol{X}_i$ in Eq. 3.13 have

a reduced number of pixels. The SLM-MACE filter is synthesized by minimizing Eq. 3.13 subject to the SLM constraints. The computation time required for the optimization will be reduced depending upon the reduction in the number of pixels of the original training images. The original filter is then recreated by placing the optimized filter pixel values in their respective positions and setting the rest of the filter pixels to zero.

The SLM-MACE filter from Test 9 (Table 4.10) was re-synthesized using this scheme. The number of training images used were three per class, and the SLM constraint used was CON2 (as in Test 9). The original training images and the filter were of size 64x64. A Threshold of 0.15E+06 (based on the D-Matrix values) was chosen for selecting the pixels for the new training images from the original ones. The number of pixels in the new training images and the filter was found to be 2122. This is nearly half of the original number of pixels (i.e. 4096). As expected the pixels which were turned 'OFF' in the original training images were from the high frequency region. The computation time taken for the synthesis of this filter was nearly 50 percent of the original program which amounts to considerable savings.

To evaluate the performance of this filter, all the images in the database were correlated with it. The results of this test are summarized in Table 4.15. The user specified correlation peaks for the training images were 0.8E+06 and 0.4E+06 for Class I and Class II, respectively. The Threshold

used for classification was 0.5E+06 (as in Test 9). Figure 4.17 shows a typical three-dimensional plot of the correlation plane for this filter. As seen in this figure the sidelobe suppression is good with a reasonably sharp peak.

As seen in Table 4.15, the distortion invariance properties of the filter are not affected by reducing the number of pixels in the training images. However there is an increase in the average correlation plane energies and a consequent decrease in the average PSR values as compared to the original filter (Test 9, Table 4.10).

Table 4.15. Correlation Plane Statistics for the SLM-MACE Filter (Constraint CON2) Constructed with Reduced Number of Pixels

Object Class	Percent Correct	Average PSR	Average Correlation Energy
mars1	100 %	3.370	3.237E+09
mars2	100 %	1.713	2.703E+09

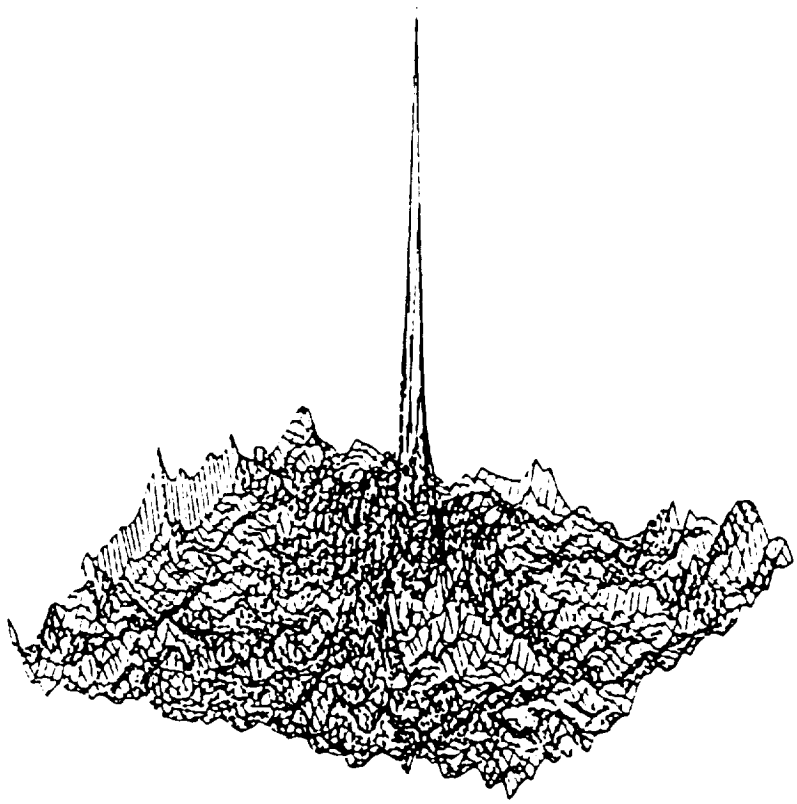

Figure 4.17. The Correlation Plane Output for SLM-MACE Filter (with Reduced Number of Pixels) Using Constraint CON2

4.5. Summary

In this chapter the distortion tolerance properties of the SLM-MACE filter were studied via computer simulations. The filter performance was compared to that of the analytical MACE, and the modified analytical MACE filters. Three different constraints were considered for the SLM-MACE filter. The SLM-MACE filter performed appreciably well in terms of its distortion invariance to the non-training images. It also performed well in the presence of noise and added bias levels in the input images. On the other hand, the modified analytical MACE filter showed poor recognition properties. A method to reduce the computation time required for the synthesis of the SLM-MACE filter was also proposed and studied.

CHAPTER 5

Summary and Recommendations

The research work leading to this report involved the design of a correlation-based distortion-invariant filter which can be used in an optical pattern recognition system. Specifically, this filter can be implemented on available spatial light modulators (SLMs) with their arbitrary realizability constraints. The SLMs considered were those exhibiting a phase-magnitude cross coupling (for example, a deformable mirror device (DMD)).

5.1. Discussion

In this report the design of a SLM-constrained minimum average correlation plane energy (MACE) filter was presented. An algorithm for its synthesis was also discussed. The SLM-MACE filters were then synthesized using three different constraints. To evaluate the performance of this SLM-MACE filter, distortion tests were carried out using computer simulations.

To date no methods have been developed to design SLM-constrained filters using purely analytical techniques. The methods proposed in the literature involve the use of certain optimization procedures. In this report the simulated annealing technique of optimization was used in the design of the SLM-MACE filter. The optimization criterion used was the

minimization of the average correlation plane energy of the filter. There were two specific constraints for this filter design. One is maintaining the user-specified values at the origin of the output correlation plane. The other is that the filter can take on only those complex values which meet the realizability constraints of a given SLM.

The distortion test results of the SLM-MACE filter were compared to that of the original analytical MACE filter and its modified version. Basically the analytical MACE filter was modified such that it would meet the SLM constraints (since it normally has arbitrary complex values which cannot be realized on available SLMs). The characteristics of the correlation planes obtained using the SLM-MACE filter were good with low sidelobes and a sharp correlation peak at the origin. However, the energy minimization of the SLM-MACE filter was not as good as the analytical MACE filter. Using the SLM-MACE filter the non-training images were correctly identified as belonging to the right class in an intra-class pattern recognition problem. This demonstrated the invariance of the filter to distortions in the input image.

On the other hand the modified analytical MACE filter response was not good with respect to the correlation plane energy minimization. Moreover it performed poorly in identifying images as belonging to a particular class. This was because the correlation peak constraints for the training images were violated caused by the modification. Thus it can be concluded that the MACE filter is not suitable for

implementation on some presently available SLMs with their limited modulating capabilities. Hence, in the future, correlation-based optical pattern recognition filter designs should take into account the realizability constraints of available SLMs.

The SLM-MACE filter performed reasonably well when the input test images were corrupted with white gaussian noise. When the test images were corrupted with a constant bias term there was no significant deterioration in its performance unlike the analytical MACE filter. However, the computation time involved in the synthesis of the SLM-MACE filter was considerable. A technique to reduce the computation time was also presented.

5.2. Recommendations

Although the SLM-MACE filter performs reasonably well and can be implemented on currently available SLMs, the synthesis procedure presented in this report has certain limitations. Also, there are certain possible extensions to this research and they are discussed in this section.

Some other optimization techniques which would require less computation time for the design of the SLM-MACE filter can be investigated. A possible variation to the SLM-MACE filter design can be the inclusion of the constraints (at the origin of the correlation plane) within the average correlation plane energy equation of the MACE filter. This way, these constraints will not have to be added to the correlation plane energy function (for example, using the

penalty function technique) as was done in this report. The energy function which includes the constraints could then be minimized using some optimization technique to determine the MACE filter such that it is constrained to an SLM operating region.

The SLM-MACE filter design concept can be extended to include the minimization of the variance in the output due to noise in the input images. Further, there is potential in pursuing the use of the SLM-MACE filter to estimate the orientation of an object as was done using the synthetic estimation filter. The design for a SLM-constrained composite filter can be modified by considering some other performance measures for optimization, for example, the peak-to-correlation energy or the signal-to-noise ratio.

REFERENCES

1. Fu, K.S., ed. Applications Of Signal Processing, CRC Press, Inc., Boca Raton, Florida, 1982, pp. 1-12.

2. Tam, E.C., Yu, F.T.S., Gregory, D.A., and Juday, R.D., "Autonomous Real-time Object Tracking with an Adaptive Joint Transform Correlator," Optical Engineering, Vol. 29, No. 4, January 1990, pp. 314-320.

3. Balendra, A., Real-Valued Composite Filters for Correlation-based Optical Pattern Recognition, M.S. Thesis, Department of Electrical Engineering, Tennessee Technological University, 1991.

4. Cassasent, D., "Pattern Recognition: a Review," IEEE Spectrum, Vol. 18, No. 3, March 1981, pp. 28-33.

5. Stark, H., ed. Applications Of Optical Fourier Transforms, Academic Press,Inc., New York, 1982.

6. Bahri, Z., and Vijayakumar, B.V.K., "Phase-Only Filter with Improved Signal to Noise Ratio," Applied Optics, Vol. 28, No. 2, January 1989, pp. 250-257.

7. Horner, J.L., "Light Utilization in Optical Correlators," Applied Optics, Vol. 21, No. 24, December 1982, pp. 4511-4514.

8. Mahalanobis, A., Vijayakumar, B.V.K., and Casasent, D., "Minimum Average Correlation Energy Filters,"Applied Optics, Vol. 26, No. 17, 1987, pp. 6633-6640.

9. Juday, R., "Correlation With a Spatial Light Modulator Having Phase and Amplitude Cross Coupling," Applied Optics, Vol. 28, No. 22, 1989, pp. 4865-4869.

10. Farn, M.W., and Goodman, J.W., "Optimal Maximum Correlation Filter for Arbitrarily Constrained Devices," Applied Optics, Vol. 28, No. 15, August 1989, pp. 3362-3366.

11. Casasent, D., "Unified Synthetic Discriminant Function Computational Formulation," Applied Optics, Vol. 23, No. 10, 1984, pp. 1620-1627.

12. Vijayakumar, B.V.K., "Tutorial Survey of Composite Filter Designs for Optical Correlators," Applied Optics, Vol. 31, August 1992, pp. 4773-4801.

13. Casasent, D., and Ravichandran, G., "Modified MACE Filters for Distortion-invariant Recognition of Mobile Targets," Proc. of SPIE, Vol. 1156, 1989, pp.177-188.

14. Sudarshanan, S.I, Mahalanobis, A., and Sundareshan, M.K., "Unified Framework for Synthesis of Synthetic Discriminant Functions with Reduced Noise Variance and Sharp Correlation Filters," Optical Engg., Vol. 29, No. 9, September 1990, pp. 1021-1028.

15. Vander Lugt, A., "Signal Detection by Complex Spatial Filtering," IEEE Trans. on Inf. Theory, IT-10, 1964, pp. 139-145.

16. Lohmann, A.W., and Paris, D.P., "Binary Fraunhofer Holograms, Generated by Computer," Applied Optics, Vol. 6, No. 10, 1967, pp. 1739-1748.

17. Ward, C., and Fisher, A.D., "Spatial Light Modulators: Applications and Functional Capabilities," Optical Signal Processing, Horner, J.L. ed., Academic Press Inc., San Diego, 1987, pp. 478-480.

18. Juday, R.D., Monroe, S.E., and Gregory, D.A., "Optical Correlation with Phase Encoding and Phase Filtering," SPIE Proc., Vol. 825, 1987, pp. 149-155.

19. Gregory, D.A., Juday, R.D., Samsell, J., Gale, R., Cohn, R.W., and Monroe, S.E., "Optical Characteristics of a Deformable Mirror Spatial Light Modulator," Optics Letters, Vol 13, No. 1, 1988, pp. 10-12.

20. Yu, F.T.S., Optical Information Processing, Wiley, New York, 1983, p. 141.

21. Horner, J.L., and Gianini, P.D., "Phase-only Matched Filtering," Applied Optics, Vol. 23, No. 812, March 1984, pp. 812-816.

22. Downie, J.D., and Reid, M.B., "Mapping Considerations for Optimal Binary Correlation Filters," Applied Optics, Vol. 29, No. 35, 1990, pp. 5235-5241.

23. Downie, J.D., "Optimizing Binary Phase and Amplitude Filters for PCE, SNR, and Discrimination," SPIE Proc., Vol. 1701, 1992, pp. 128-139.

24. Juday, R.D., and Daiuto, B.J., "Relaxation Method of Compensation in an Optical Correlator," Optical Engg., Vol. 26, No. 11, 1987, pp. 1094-1101.

25. Horner, J.L, and Gianino, P.D., "Applying the Phase-only Concept to the Synthetic Discriminant Function Correlation Filter," Applied Optics, Vol. 24, No. 6, March 1985, pp. 851-854.

26. Kallman, R.R., "Direct Construction of Phase-only Filters," Applied Optics, Vol. 26, No. 4, December 1987, pp. 5200-5201.

27. Jared, D.A., and Ennis, D.J., "Inclusion of Filter Modulation in Synthetic-discriminant Function Construction," Applied Optics, Vol. 28, No. 2, January 1989, pp. 232-239.

28. Balendra, A., and Rajan, P.K., "Real-valued Composite Filters for Optical Pattern Recognition," Proc. SPIE Syst. Arch. Conf., San Diego, 1992.

29. Carlson, D.W., and Vijayakumar, B.V.K., "Design of Correlation Filters for Arbitrarily Constrained Devices," Final Report for Grant NAG-9-523, 1992.

30. Vijayakumar, B.V.K., and Hassebrook, L., "Performance Measures for Correlation Filters," Applied Optics, Vol. 29, No. 20, 1990, pp. 2997-30006.

31. Fletcher, R., and Powell, M.J.D., "A Rapidly Convergent Descent Method for Minimization," Computer J., Vol. 6, 1963-1964, pp. 163-168.

32. Hooke, R. and Jeeves, T.A., "Direct Search Solution of Numerical and Statistical Problems," J. Assoc. Comp., Mach., Vol. 8, 1961, pp. 212-229.

33. Kirkpatrick, S., Gelatt, C.D., and Vecchi, M.P., "Optimization by Simulated Annealing," Science, Vol. 220, No. 4598, May 1983, pp. 671-680.

34. Carnevali, P., Coletti, L., and Patarnello, S., "Image Processing By Simulated Annealing," IBM J. Res. Develop., Vol. 29, No. 6, November 1985, pp. 569-579.

35. Kim, M.S., and Guest, C.C., "Simulated Annealing Algorithm For Binary Phase-only Filters in Pattern Classification," Applied Optics, Vol 29, No. 8, March 1990, pp. 1203-1208.

36. Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., and Teller, E., "Equation of state calculations by fast computing machines," J. Chem. Phys., Vol. 21, No. 6, 1953, pp. 1087-1090.

37. Mahalanobis, A., and Casasent, D.P., "Performance Evaluation of Minimum Average Correlation Filters," Applied Optics, Vol. 30, No. 5, 1991, pp. 561-572.

38. Arsenault, H. et al., "Modified Composite Filters for Pattern Recognition in the Presence of Noise with a Non-zero Mean," Optical Commun., Vol. 63, 1987, pp. 15-20.

39. Bahri, Z., Phase-only and Binary Phase-only Filters in Optical Correlators, Ph.D. Dissertation (Chapter 2), Dept. of Electrical and Computer Engineering, Carnegie-Mellon University, 1989.

APPENDIX

```
C
C                    PROGRAM SLM_MACE.FOR
C
C
C This program solves the SLM-constrained MACE filter problem
C using the simulated annealing method of optimization. The
C number of variables in the optimization are L = N*N (The
C size of the training images and the filter being NxN).
C
C A Total of upto 10 training images can be used in this
C program, i.e. 5 for Class I, and 5 for Class II.
C
C 'COMPUTEFN' is the routine which evaluates the objective
C function at the given variable values.
C
C
C ************************************************************
C
C Definitions of some important variable names used.
C ..........................................................
C
C N,M == NxN or MxM are the Image/Filter dimensions.
C
C L == Total no. of pixels in the Images ( i.e. = NxN).
C
C KKK == Number of training Images.
C
C iter == Maximum number of iterations permitted.
C
C ased,bsed,sed == Contain the seed values for use in the
C    random number generation function. These seed values are
C    generated using the 'secnds' function in FORTRAN.
C
C val1, val2 == User-specified peak-values for Class I and
C    Class II resp.
C
C k1, k2, k3 == Constants used in the Penalty functions for
C    the constraints of the objective function.
C
C XX == Array of dimension LxKKK. The adjacent columns of this
C    array contain the real and imaginary parts of the
C    training Image vectors.
C
C dmat == Array of dimension L. Contains the average of the
C    (magnitude)**2 of all the training Image vectors.
```

```
C cst == Array of dimension Lx2. Contains the SLM-contraints
C    in two columns representing the phase and magitude
C    respectively.
C
C delta == Size of the perturbations of the phase of the
C    filter pixels. delta = (2*π)/(total number of samples)
C
C tempx1 == Array of dimension L. It contains the temporary
C    values of the phases of the filter pixels during
C    optimization.
C
C X == Array of dimension L. It will contain the final phase
C    values of the pixels of the optimized filter.
C
C objf == Contains the value of the objective function
C    resulting from a successful perturbation of a variable
C    during an iteration of the optimization process. The
C    function 'COMPUTEFN' also returns the value of the
C    objective function in 'objf'. At the end of the
C    optimization 'objf' will contain the final objective
C    function value.
C
C FF == Dimension 2. Used to store the temporary values of the
C    objective function as a result of a perturbation.
C
C objo == Contains the value of the average correlation plane
C    energy.
C
C nobjo == Used to store the temporary value of the
C    correlation plane energy as a result of the perturbation
C    of filter pixel phase value.
C
C con == Array of dimension KKKx2 .Holds the real and
C    imaginary parts of the correlation peak constraints as
C    the two entries of each row.
C
C ncon== Array of dimension KKKx2. Holds the temporary values
C    of the above constraints.
C
C minX == array of dimension L. (see below for explanation)
C
C minobjf == Contains the least value of the objective
C    function obtained during the optimization. After every
C    iteration, the value in 'objf' is compared to 'minobjf'.
C    The smaller of the two values is retained in 'minobjf'.
C    The corresponding phase values of the filter pixels are
C    stored in the array 'minX'. This is done by calling
C    SUBROUTINE REC.
C
C*************************************************************
```

```
      integer*2 N
      integer mm,oncem,NN,M,L,KKK,iter,N0,dev1
      integer sed,ased,bsed,pp,count
      integer txn,xn,accept,attempt
      real sed1,sed2,sed3,minobjf,thi,nobjo,nobjo1
      parameter(M=64)
      parameter(L=4096)
      parameter(KKK=4)
      integer*2 ipint(M,M)
      real tempx1(L),X(L),dmat(L),FF(2),minX(L),ncon(KKK,2)
      real ipr(M,M),XX(L,2*KKK),cst(1:4096,2),con(KKK,2)
      real val1,val2,k1,k2,k3,rlfil,imfil,temp0
      real randnum1,randnum2,delta,prob,temp,phase
      real objo,objf,xi,xr,txi,txr,pai,oldf,perc
      complex ipc(M,M),ipf(M,M),filt(L),filt1(M,M)
      character filename1*14,filename2*14,filename3*14
      character filename4*14

      common N

      N=M
      dev1=6
      sed1=40000.0
      sed2=50000.0
      sed3=90000.0
      pp = 0
      accept = 0
      attempt = 0
      NN = 0
      oncem=0
      mm=0
      pai=3.141592654


C Generate the seed values for use in the random no.
C generation function 'RAN'

      ased=abs(2*(int(secnds(sed1)))+1)
      sed=abs(2*(int(secnds(sed2)))+1)
      bsed=abs(2*(int(secnds(sed3)))+1)
      iter = 10000
      val1=800000.0
      val2=400000.0
      k1=100.0
      k2=200.0
      k3=50.0
      delta=3.0691964E-03
```

```
C 'images4.na' contains the names of the training images which
C will be read one after another. The real and imaginary part
C of the fourier transformed images are stored as columns of
C the array XX. The magnitude squared values of the fourier
C transformed image pixels are computed and stored in 'dmat'.

        filename1='images4.na'
        open(unit=11,file=filename1,status='old')
 32     read(11,1,end=900)filename2
 1      format(a12)
        call getint1(ipint,filename2)
        call int_rl(ipint,ipr)
        do j=1,N
           do i=1,N
              ipc(i,j)=cmplx(ipr(i,j))
           end do
        end do
        call tdfft(1.0,ipc,ipf)
        mm=mm+1
        jj=0
        do i=1,N
           do j=1,N
              jj=jj+1
              XX(jj,mm) = real(ipf(i,j))
              XX(jj,mm+1) = aimag(ipf(i,j))
              dmat(jj) = dmat(jj)+(XX(jj,mm)**2)+
     $                   (XX(jj,mm+1)**2)
           end do
        end do
        mm=mm+1

C Get the next training image.

        go to 32

C Take the average of all the elements of 'dmat'.

 900    do i=1,L
           dmat(i)=dmat(i)/float(KKK)
        end do


C Initialize the array which will contain the SLM-constraints
C table.

       do i=1,4096
          do j=1,2
            cst(i,j)=0.0
          end do
       end do
```

```
C Read the SLM-constraint data file into the array 'cst'.

      open(unit=11,file='cstraintn.in2048',status='old')

      do i=1,2048
         read(11,*)(cst(i,j),j=1,2)
      end do

      close(unit=11)


C Determine the starting points for the optimization. Here the
C starting phase values for the filter pixels are those
C obtained from the previously solved Analytical MACE filter
C which was stored in the file "filt226.s64".
C 'GETCMP' is an unformatted-read  routine which reads the
C complex filter into an array 'filt1'.

      filename4='filt224.s64'
      call getcmp(filt1,filename4)
      ii=0
      do i=1,N
         do j=1,N
          ii=ii+1
          phase=atan2(aimag(filt1(i,j)),real(filt1(i,j)))
          if(phase.lt.0.0)then
            phase=phase+(2.0*pai)
          end if
          tempx1(ii)=cst(nint(phase/delta),1)
         end do
       end do


C Initial Guess. This is another way of determining the
C starting points for the phase values of the filter pixels
C for optimization. Here the initial phase values are selected
C randomly from the SLM-constraints table.
C
C       do i=1,L
C         N1=int(2048.0*ran(sed))
C         tempx1(i)=cst(N1,1)
C    end do


C Evaluate the cost function with the initial guess. The
C routine 'COMPUTEFN'evaluates the objective function value at
C the given points of the filter pixel values.

        do i=1,L
          X(i)=tempx1(i)
        end do
 420    call computefn(NN,L,KKK,XX,X,dmat,cst,k1,k2,k3,
     $                 delta,val1,val2,objo,objf,con)
```

```
      if(oncem.ne.0)then
         perc=((oldf-objo)/oldf)*100
         oldf=objo
         if(perc.le.(2.0))then
            count=count+1
         else
            count=0
         end if
      else
         oldf=objo
      end if

C The initial value of the temperature 'temp0' is got from the
C starting value of the objective function 'objf'.
C 'temp0' does not change during the entire run of the
C optimization. The temperature parameter which is updated as
C the iterations progress is 'temp'. Here some data files are
C also created.

      FF(1)=objf
      temp0=objf/2.0
      temp=temp0
      call rec(L,X,minX,objf,minobjf)
      write(25,*)0,objf
      write(26,*)0,objo
      write(27,*)oncem,objf
      write(28,*)oncem,objo
      if(count.ge.3)go to 500
      oncem=oncem+1


C ****    The optimization iterations begin here   **********

      write(dev1,*)' PERFORMING OPTIMIZATION !! PLEASE WAIT'
      write(dev1,*)'  '

      do i=1,iter
         do j=1,L
            pp=pp+1


C Generate a random no. to determine whether a positive or
C negative perturbation be caused for each variable.

                  randnum1=ran(ased)
                  if(randnum1.le.0.5)then
                     X(j)=tempx1(j)+(delta)
                     if((nint(X(j)/delta)).gt.2048)then
                        X(j)=delta
                     end if
                  else
                     X(j)=tempx1(j)-(delta)
                     if((nint(X(j)/delta)).lt.1)then
```

```
                X(j)=2.0*pai
              end if
            end if

C The new objective function value due to the perturbation is
C computed here and its value is stored in FF(2). The old
C value of the objective function is stored in FF(1).
C Note: Here the entire objective function is not evaluated.
C Rather the change in the old objective function value caused
C by the perturbation is computed.

            txn=nint(tempx1(j)/delta)
            txr=cst(txn,2)*cos(cst(txn,1))
            txi=cst(txn,2)*sin(cst(txn,1))
            xn=nint(X(j)/delta)
            xr=cst(xn,2)*cos(cst(xn,1))
            xi=cst(xn,2)*sin(cst(xn,1))
            nobjo=objo-(dmat(j)*(txr**2+txi**2))+(dmat(j)*
     $           (xr**2+xi**2))
            mm=0
            do ii=1,KKK
               mm=mm+1
               ncon(ii,1)=con(ii,1)+(XX(j,mm)*(xr-txr))
     $                    +(XX(j,mm+1)*(xi-txi))
               ncon(ii,2)=con(ii,2)+(XX(j,mm)*(xi-txi))
     $                     +(XX(j,mm+1)*(txr-xr))
               mm=mm+1
            end do
            nobjo1=0.0
            jj=0
            kk=KKK/2
            do ii=1,KKK-1,2
              jj=jj+1
              kk=kk+1
              nobjo1=nobjo1+(k3*((ncon(jj,1)-val1)**2))+
     $         (k1*((ncon(kk,1)-val2)**2)) +
     $        (k2*(ncon(ii,2)**2))+(k2*(ncon(ii+1,2)**2))
            end do
            FF(2)=nobjo+nobjo1

C If the new objective function value (i.e. due to the
C perturbation) is smaller than the current value then the
C perturbation is accepted, and the objective function value
C 'objf' is updated along with all the variables used to
C to compute it. For an accepted perturbation the counter
C 'accept' is incremented.
```

```
                if(FF(2).lt.FF(1))then
                   FF(1)=FF(2)
                   objo=nobjo
                   do ii=1,KKK
                      con(ii,1)=ncon(ii,1)
                      con(ii,2)=ncon(ii,2)
                   end do
                   objf=FF(2)
                   tempx1(j)=X(j)
                   accept=accept+1
                   go to 701
                end if

C If the perturbation results in an increase in the objective
C function value then, it will only be accepted with a certain
C probability, computed in the variable 'prob' below. Again if
C this perturbation is accepted then the objective function
C value 'objf' is updated along with the variables used to
C compute it.

                thi=(FF(2)-FF(1))/temp
                if(thi.ge.80.0)then
                   thi=80.0
                elseif(thi.le.0.000001)then
                   thi=0.0
                end if
                prob=1.0/(1.0+exp(thi))
                randnum2=ran(bsed)
                if(randnum2.lt.prob)then
                   FF(1)=FF(2)
                   objo=nobjo
                   do ii=1,KKK
                      con(ii,1)=ncon(ii,1)
                      con(ii,2)=ncon(ii,2)
                   end do
                   objf=FF(2)
                   tempx1(j)=X(j)
                   accept=accept+1
                   go to 701
                end if

C If the perturbation is not accepted then the filter pixel
C value is restored to its previous value.

                X(j)=tempx1(j)
                if(objf.lt.minobjf)
701                rec(L,X,minX,objf,minobjf)
     $
             end do

             write(25,*)i,objf
             write(26,*)i,objo
```

```
C ****      Updating the Temperature parameter    ************
C
C If the no. of accepted perturbations >= 10xL then update the
C temperature parameter 'temp', and proceed. Initialize the
C counter 'attempt', 'accept'.

            if(accept.ge.10*L)then
               temp=((0.98)**i)*temp0
               accept=0
               pp=0
               attempt=0
            end if


C If the no. of attempted perturbations (without acceptance)
C is >= 100xL then update the temperature parameter and make
C a note of this fact by incrementing the counter
C 'attempt'.

            if(pp.ge.100*L)then
               temp=((0.98)**i)*temp0
               accept=0
               pp=0
               attempt=attempt+1
            end if


C If the above has occurred thrice continuously, then
C terminate the optimization. This is the convergence test. If
C so then branch to 200.

            if(attempt.ge.3)then
               attempt=0
               accept=0
               go to 200
            end if

            if(temp.le.(1.0E-04))then
               attempt=0
               accept=0
               go to 200
            end if

 779     end do


C If the convergence criterion is not met after 'iter'
C iterations, then quit the program. ( This so that the
C program does not get into an endless loop).


      write(dev1,*)'    '
      write(dev1,*)' **************************** '
```

```
      write(dev1,*)' ERROR ERROR ERROR '
      write(dev1,*)' ****************************** '
      write(dev1,*)'  '
      write(dev1,*)' No convergence after ',i,' iterations'
      go to 300
200   write(dev1,*)
      write(dev1,*)' ****************************** '
      write(dev1,*)' Converged after ',i,' iterations'


C 'Minobjf' contains the least (minimum) value of the
C objective function encountered during the optimization. The
C phase values corresponding to this minimum value are stored
C in the array 'minX' (of dimension L). If the final value of
C the objective function (at the end of the optimization) is
C not the least then the least value is taken.


 300    if(objf.gt.minobjf)then
           do zz=1,L
              X(zz)=minX(zz)
           end do
        end if
        callcomputefn(NN,L,KKK,XX,X,dmat,cst,k1,k2,k3,delta,
     $       val1,val2,objo,objf,con)
       write(dev1,*)'  '
       write(dev1,*)' The temp is',temp
       write(dev1,*)'  '


C The optimization is repeated with increased values of the
C constraints penalty violation constants 'K1', 'K2', 'K3'.

        if(oncem.ne.20)then
           k1=k1+6.0
           k2=k2+12.0
           k3=k3+3.0
           write(27,*)oncem,objf
           write(28,*)oncem,objo
           go to 420
        end if


C Using the phase values of the filter pixels from the array
C 'X' pick out their corresponding magnitude values and
C compute the complex filter pixel values. Store these complex
C filter pixel values in the array 'filt'.
```

```
500     do i=1,L
           N0=nint(X(i)/delta)
           rlfil=cst(N0,2)*cos(cst(N0,1))
           imfil=cst(N0,2)*sin(cst(N0,1))
           filt(i)=cmplx(rlfil,imfil)
        end do

C Arrange the filter from a vector form into a matrix form.

        jj=0
        do i=1,N
           do j=1,N
              jj=jj+1
              filt1(i,j)=filt(jj)
           end do
        end do

C  Save the final filter in a file for future correlations.

        filename3='filtthesis.s64'
        open(unit=14,file=filename3,form='unformatted'
     $       ,status='unknown')
        write(14)filt1
        close(unit=14)

C  Write the final value of the objective function and
C  constraints.

        write(dev1,*)'Total Objective fn. value is',objf
        write(dev1,*)'The Correlation Energy is',objo
        write(dev1,*)'Constants k1,k2,k3 are',k1,k2,k3
        write(dev1,*)'The Constraints (real,imag) are:'
        do i=1,KKK
           write(dev1,*)con(i,1),con(i,2)
        end do

      stop
      end
```

```
      subroutine rec(L,X,minX,objf,minobjf)

C ***************************************************************
C
C (input) L == Dimension of the array X defined in the main
C              program above.
C
C (input) X == Array of dimension L containing the current
C    phase values of the filter pixels.
C
C (input) objf == Contains the value of the objective function
C    evaluated from the array X.
C
C (output) minX == Array of dimension L. The values of X are
C    stored into 'minX'
C
C (output) minobjf == The value from 'objf' is stored into
C    'minobjf'
C
C***************************************************************
      real X(L),minX(L),objf,minobjf

      do i=1,L
         minX(i)=X(i)
      end do
      minobjf=objf

      return
      end


      subroutine computefn(NN,L,KKK,XX,X,dmat,cst,k1,k2,k3,
     $                     delta,val1,val2,objo,objf,con)

C ***************************************************************
C
C A routine to calculate the value of the objective function,
C the value of the correlation energy plane, and value of the
C constraints. XX is a 2 dimensional array containing the real
C part of the elements of the reference image in the first
C column and the imaginary parts of the elements in the second
C column. X is a one dimensional array. The elements of this
C array correspond to the phase values of the filter pixels.
C
C
C NN (input)
C
C L  (input)
```

```
C XX (input array) == Each of the columns correspond to the
C     real and imaginary part of the fourier transformed
C     images.
C
C X  (input array) == Contains the phase values of the filter
C     pixels.
C
C cst (input array) == Contains the SLM constraint table.
C
C K1, K2, K3 (input) == as defined in the main program.
C
C delta (input) == as defined in the main program.
C
C val1, val2 (input) == as defined in the main program.
C
C objo,objf (output) == correlation plane energy value, and
C     the objective function value respectively.
C
C con (output array) == Correlation plane constraints.
C
C ***************************************************************
      integer NN,L,KKK,num,i,j,jj
      real XX(L,20),X(L),cst(4096,2),k1,k2,k3
      real objo,objf,objo1,delta,con(KKK,2),dmat(L)
      real rlf,imf,ave,val1,val2

      objo=0.0
      objo1=0.0
      objf=0.0
      ave=0.0


      do i=1,KKK
         con(i,1)=0.0
         con(i,2)=0.0
      end do

      do j=1,L
        num=nint(X(j)/delta)
        rlf=cst(num,2)*cos(cst(num,1))
        imf=cst(num,2)*sin(cst(num,1))
        objo=objo + (dmat(j)*(rlf**2+imf**2))
        mm=0
        do i=1,KKK
           mm=mm+1
           con(i,1)=con(i,1) +
     $              ((XX(j,mm)*rlf)+(XX(j,mm+1)*imf))
           con(i,2)=con(i,2) +
     $              ((XX(j,mm)*imf)-(XX(j,mm+1)*rlf))
           mm=mm+1
        end do
      end do
```

```
      objo1=0.0
      j=0
      jj=KKK/2
      do i=1,KKK-1,2
         j=j+1
         jj=jj+1
         objo1=objo1 + (k3*((con(j,1)-val1)**2))+
     $          (k1*((con(jj,1)-val2)**2))+
     $          (k2*(con(i,2)**2))+(k2*(con(i+1,2)**2))
      end do
      objf=objo + objo1

      return
      end



      subroutine getint1(array,arrayname)

C ***************************************************************
C
C A routine to read an integer N by N array stored in an
C 'unformatted' format.
C
C
C arrayname (input) == Name of file containing the array
C array (output) == Integer array from the file arrayname
C
C ***************************************************************

      common N
      integer*2 array(N,N),N
      character*14 arrayname

      open(unit=8,file=arrayname,form='unformatted',
     $     status='old')

      print *,' '
      print *,'Reading the file '//arrayname
      read(8) array
      close(unit=8)

      print *,'File transfer completed.'
      print *,' '

      return
      end
```

```
      subroutine getcmp(array,arrayname)
C ***************************************************************
C
C  A routine to read a complex  N by N array stored in an
C  'unformatted' format.
C
C
C  arrayname (input) == Name of file containing the array
C  array (output) == Array from the file arrayname
C
C***************************************************************

      common N
      integer*2 N
      complex array(N,N)
      character*14 arrayname

      open(unit=8,file=arrayname,form='unformatted',
     $    status='old')

      print *,' '
      print *,'Reading the file '//arrayname
      read(8) array
      close(unit=8)

      print *,'File transfer completed.'
      print *,' '

      return
      end
```